

What's New for PC ARC/INFO 4.0

This guide is primarily intended for existing users of PC ARC/INFO. New users will find this discussion useful, but are recommended to refer to the documentation that accompany this release. They introduce the concepts of PC ARC/INFO. The on-line Help includes a 'Discussion Topics' section which provides information on starting and using PC ARC/INFO 4.0 as well as Command Reference sections which detail the use of each command.

Highlights of PC ARC/INFO 4.0

- Windows 32 bit Application
- Double Precision Coverages
- Background Images in ARCEDIT and ARCPLOT
- ARC commands available in all modules
- TABLES subcommands replaced with ARC processor commands
- Permanent Relates
- Shared Selection Sets between modules
- New functionality
- Support for Annotation subclasses and stacked annotation
- WinTab digitizer support
- Improved Customization tools
- Increased limits
- Improved performance
- Faster searches - Indexed items
- Improved menu interface

Contents:

Directory and command processor changes

- Environment variables are no longer required
- New ARC command processor
- New ARCEXE directory structure
- Implementation of External and Internal commands
 - Most ARC commands are accessible from all modules
 - COMMANDS displays both Internal and External commands
- New command search path
- New SML search path
- Custom commands
- ARC has optional {/w} parameter on startup
- New names for PC ARC/INFO Windows menus
 - Updated menu interface
- New module initialization files

Contents cont.

Changing workspace and drive location with &WS, A and CD
New directory name for temporary ARC files
WKSP directory now includes a list of installed fonts
On-line Help file has a new organization

Commands that have been removed or replaced

ROUTE and ALLOCATE have been removed
The following PC ARC/INFO commands have been removed or replaced
 These ARC commands have been removed
 These ARC commands have been replaced
 These ARCPLOT commands have been removed
 These ARCEDIT and ARCPLOT commands have been replaced
 These SML Developer Toolkit macros have been removed
 These SML Developer Toolkit macros have been replaced
TABLES subcommands have been replaced with ARC level commands
Usage changes for existing commands
 ARC usage changes
 ARCEDIT usage changes
 ARCPLOT usage changes
New commands
 New Global commands
 New ARC commands
 New GENERATE subcommands
 New ARCEDIT commands
 New ARCPLOT commands
 New commands for ARCEDIT and ARCPLOT
 New SML commands
 New SML Developer's Toolkit commands

PC ARC/INFO 4.0 Limits

Converting PC ARC/INFO Single Precision coverages to Double Precision

Conversion Issues
These commands will work with both Single and Double Precision Coverages
Commands that Convert between Single Precision and Double Precision Coverages
Using COPYCOV and COPYWKSP
Coordinate Shift Issues
OEM / ANSI / ISO International Character Issues
Converting TEXT and DATABASE files used by PC ARC/INFO
Some coverage file names have changes
Some default item definitions have changes
What happens to Address and Geocoding files when a coverage is converted to
 PC ARC/INFO 4.0?
What happens to single precision annotation now that PC ARC/INFO 4.0 supports annotation
 subclasses?
What happens to PC ARC/INFO 4.0 annotation when it is converted to single precision?
A note about annotation that is brought in to PC ARC/INFO 4.0 with User Position and User
 Offset characteristics

New Windows TABLES command

Managing data files with the new ARC level database management commands

- Managing Data files
- Manipulating Records
- Manipulating Items

There is a usage change for [ARC] LIST

Continued support for concatenated items and Item ranges

Item ranges can be processed in reverse order

New L command options

LISTCOVS supports both single and double precision coverages

Selection Sets can be created at ARC as well as in ARCEDIT and ARCPLOT

Selection Sets can be used between PC ARC/INFO modules and sessions

The **RELATE** command replaces the **TABLES JOIN** command

- Permanent multilevel RELATES
- RELEDIT - A Windows interface for RELATE
- INFODIRECTORY will list the RELATE status of each of the database files

ITEMINDEX creates a new index file for indexed items

- Indexed items are now automatically used with certain simple logical expressions
- INDEXEDIT is a new item index file management tool
- KILLINDEX can be used to remove an item index file or the spatial index file

New file format for the output from UNLOADANNO

LOADANNO supports annotation subclasses

You can now specify a background color for all text windows using the **COLOR** and **DAILOGCOLOR** commands

DIALOGCLEAR has been added to the ARC module

PC ARC/INFO command lines can include up to 1024 characters

WinTab and **DIGFORM** digitizer support

The following commands now display Date items using 4 digits instead of 2

STATISTICS command at [ARC]

CLASSIFY is a new interactive version of the **CLASS** command in ARCPLOT

New **TIGERTOOL** command

New **SHAPEDXF** command

Quoted strings on command lines can be enclosed with either single or double quotation marks

All 'External' commands now return an error code to indicate successful or unsuccessful command processing

The results of an &SYSTEM program can also be retrieved with the &RV statement

SML Enhancements

Complex logical expressions are now supported in SML
The block-structured &IF statement can now be used in SML and CML macros
A new '&IFNOT' conditional statement has been added to SML and CML logical processing
Many new logical operators have been added to SML and CML logical processing
New Arithmetic Operators
New SML commands
Named variables are now supported in SML by specifying them with the new &VAR statement
The Arc WIN command has a new SML companion command named &WIN
The command &WIN FILE has a new parameter controlling the length of the returned pathname as well as revised options for listing files and coverages
&WIN IDLE now has a {seconds} option
&WIN RUN / RUNW return new values upon command completion
There are now more &WIN DB Dialog Definition File Statements and Dialog File Commands.
These add more user control to the look of on-screen dialog boxes.
Dialog File Commands
Dialog File Statements
Number of local variables increased to 50
SML variables can include strings up to 256 characters
Edit boxes can have input strings up to 256 characters
&LISTVAR will now list variable values in either ascending or descending numeric order (including the new SML named variables). It can also provide information about named variables
&R is a new version of &RUN
Developer's macros now use the new &R directive to execute. &RUN and @ will no longer work with them
&KEY has a new usage
&ASK and &RESPONSE can display up to 1024 characters
In order to maintain numeric precision, the results of some calculations performed by &CV will be stored in scientific notation

Changes affecting both ARCPLOT and ARCEDIT

It is now possible to pass command parameters into ARCPLOT and ARCEDIT on start-up
DISPLAY no longer clears the text screen
SCREENSAVE creates a new display list file format
MEMO items no longer supported
SYMEDIT - a new interactive Symbol Editor
SYMPICK - a new interactive symbol picker
Menu options available when using the mouse
New symbol file options for LINESET / MARKERSET / TEXTSET / SHADESET allow symbol files to be altered and created
Saving symbol files: LINESAVE / MARKERSAVE / TEXTSAVE / SHADESAVE
SHOW LINEINFO / MARKERINFO / TEXTINFO / SHADEINFO are new commands
IMAGE support in ARCEDIT and ARCPLOT
IMAGES in ARCEDIT
IMAGES in ARCPLOT
Use the SHOWDIB command to preview images

Highlights of ARCEDIT 4.0

Continual cursor tracking when digitizing

- Annotation in ARCEDIT
 - Annotation subclasses
 - Stacked annotation in ARCEDIT
 - ANNOALIGNMENT establishes how blocks of text or stacked annotation will be drawn
 - New annotation pseudo items
- New label-point pseudo items
- Arc and Label attributes can be written and read with the PUT and GET commands
- ARCPLLOT commands can be used in ARCEDIT

Highlights of ARCPLLOT 4.0

- Enhanced printing support in ARCPLLOT
- Feature on Feature selection in ARCPLLOT
- Feature selection based on the location of a point is now supported
- Defining lines and polygon outlines in ARCPLLOT with such commands as LINE and SHADE
 - allow additional controls as splining, squaring and undoing the last point
- Making your own fonts, characters and marker symbols
- Using bitmap markers in ARCPLLOT and ARCEDIT
- Size and position of drawing elements can be interactively altered
- MAP compositions can be opened and drawn on top of existing graphics
- SHOW LINESET / MARKERSET / SHADESET / TEXTSET are new in ARCPLLOT
- New ARCPLLOT annotation and text features
 - Stacked and concatenated text strings in ARCPLLOT
 - TEXTALIGNMENT - new command in ARCPLLOT

Directory and command processor changes

Environment variables are no longer required

Environment variables are no longer required for Startup. You can also omit the Path statement to ARCEXE\CMD in the AUTOEXEC.BAT file.

However, the following variables are available for use with the SML function &VALUE:

ARC	Path name of the ARCEXE directory. This is the directory where the files needed to run PC ARC/INFO reside.
ARCPROG	The PC ARC/INFO program that is currently running. It returns 'ARC', 'ARCPLLOT' or 'ARCEDIT'. <i>This is a new environment variable.</i>
SML	Path name of the SML directory (SMLDIR command). It is one of the places ARC looks for files when you use &R or &RUN.
WKSP	Path to the WKSP directory for this session. This is where temporary files and coverages are kept. It is different for each session. It always ends with a "\".
WKSPDIR	Path to the directory of all WKSP directories. Users can set this before starting PC ARC/INFO or Windows.

New ARC command processor

Command processing at version 4.0 is significantly different from all 3.x versions of PC ARC/INFO. This is because commands are now executed as CML/SML macros rather than from DOS batch files (.BAT files) and as DOS executables.

New ARCEXE directory structure

Converting ARC commands from DOS executables to CML/SML macros meant the PC ARC/INFO directory structure had to change. (A CML is a compiled SML.) The CPLLIB directory is no longer required and has been removed. The PTOOL subdirectory now contains the command CML files rather than the DOS BATCH files (organized by module) and the UTOOL directory has moved under the PTOOL directory (since it also contains SMLs and CMLs). The UTOOL directory is no longer a place only for custom macros but is now a directory that includes commands that can be executed universally from ARC, ARCEDIT and ARCPLOT.

ARCEXE - \CMD (startup command and module icon files)
 \DATUM (datum files used by PROJECT)
 \DIGFORM (digitizer format files)
 \HELP (HELP source files)
 \IGL (IGLFNT)
 \INSTALL
 \LOCALE (annotation conversion files for ISO/ANSI and ANSI/ISO conversion)
 \MENU (ARCED.DBF)
 \PROGRAMS (command executables)
 \PTOOL
 \ARC (ARC PROCESSOR files and CML/SMLs)
 \ARCEDIT (ARCEDIT only files and CML/SMLs)
 \ARCPLOT (ARCPLOT only files and CML/SMLs)
 \UTOOL (External files and CML/SMLs shared by all modules)
 \RAINBOW (Hardware key drivers for International versions of PC ARC/INFO)
 \SRC (source SMLs, SML examples, PC ARC/INFO tools)
 \SYMBOLS (symbol sets)
 \TEMPLATE (PC ARC/INFO template files)
 \VOCAB (PC ARC/INFO vocabulary files)

Implementation of External and Internal commands

Commands that are stored in the ARCEXE\PROGRAMS directory are called Internal commands. All commands that are stored on the PTOOL subdirectory are called External commands.

Most ARC commands are accessible from all modules

PC ARC/INFO commands are now executed as CML and SML macros. (A CML is a compiled SML). Since each module has an SML processor, many of these CML/SMLs can be run universally from ARC, ARCEDIT and ARCPLOT. Commands that perform overlay operations and data conversions are examples of commands that are not module specific. They are stored in the PTOOL\UTOOL directory and can now be run from ARCEDIT and ARCPLOT, as well as from ARC. Some commands are specific to a module's function are not shared. The PTOOL\module directories contain these commands.

For example, commands such as POLYGONSHADES and MAPUNITS are cartographic tools and are Internal to ARCPLOT because they mean nothing to ARCEDIT or ARC. The EDITCOVERAGE command is Internal to ARCEDIT. However, many programs such as those

that perform coverage management, topology building, overlay operations or data conversions are not restricted to a specific module. They reside in the PTOOL\UTOOL directory and are available for use at any time during a PC ARC/INFO session.

COMMANDS displays both Internal and External commands

As described above, now that there are two types of commands, the COMMANDS command differentiates between Internal and External commands in its display. For example:

[ARCEDIT] **COMMANDS M**

```
-----INTERNAL COMMANDS-----
MAPEXTENT      MARKER      MARKERSAVE      MARKERSET
MARKINDEX      MOVE       MOVEITEM
-----EXTERNAL COMMANDS-----
MAPJOIN        MD         MENU            MIADSARC
MIFSHAPE       MKDIR     MNODE           MODITEM
MOSSARC
```

New command search path

- 1) Internal commands - Established on module startup by PC ARC/INFO
- 2) ARCEXE\PTOOL\module_subdir (i.e., whichever is current: ARC, ARCEdit, ARCPLOT)
- 3) ARCEXE\PTOOL\UTOOL

New SML search path

Since the directory structure for PC ARC/INFO has changed, ARCEXE\UTOOL no longer exists. This is the new SML search path:

- a) current directory
- b) SMLDIR
- c) ARCEXE\PTOOL\module (*module specific command*)
- d) ARCEXE\PTOOL\UTOOL (*global command*)

Custom macros saved in the current directory or in the directory specified by SMLDIR require the &R or &RUN directive to execute them. Use them the same way as with previous versions of PC ARC/INFO. These two paths have not changed. Only those SMLs originally saved in ARCEXE\UTOOL need to be moved. Refer to the SMLDIR command reference if you have not used an SML directory before.

Note that &RUN remains an SML directive because many User applications still exist that use only global variables. In practice, new applications should pass values between routines through local variables with the &R directive.

Custom commands

Now that PC ARC/INFO commands are compiled CML/SMLs, it is possible to load a custom SML or CML as a command in PC ARC/INFO. Copying your SML to PTOOL\module would make it available as a command at that module's prompt (ARC, ARCEdit or ARCPLOT). Saving it in PTOOL\UTOOL would make it universally available throughout a PC ARC/INFO session. These SMLs run without issuing the &R directive.

You may add your own commands to the PTOOL subdirectories, but do not modify or rename existing PC ARC/INFO commands as this may cause PC ARC/INFO to fail.

ARC has optional `/w` parameter on startup that will cause the text window to 'wait' for confirmation before exiting the ARC processor.

ARC `{/w {command {command_arguments} / sml_file}}`

When ARC starts with the `/W` parameter in use, `{command}` or `{sml_file}` is automatically executed. After `{command}` is run, the message "Press Enter to Close..." appears on the screen. The ARC processor will quit only after the user pushes ENTER.

New names for PC ARC/INFO Windows menus

To start the ARC menu use the command ARCMENU. When in ARCEDIT use EDITMENU and to start the ARCPLOT menu enter PLOTMENU. Typing MENU at any PC ARC/INFO prompt will initiate that particular module's Windows menu. Because all PC ARC/INFO commands are now SMLs, you do not need to include the `&R (@)` directive.

Updated menu interface

All menu interfaces for PC ARC/INFO 4.0 have improved in their 'look' and functionality.

New module initialization files

The ARC processor will no longer run a default macro named *module.SML*.

If you created and used a file named ARC.SML, ARCEDIT.SML and/or ARCPLOT.SML that would automatically be run when ARC, ARCEDIT or ARCPLOT was started, you will have to rename it. This is because command names are now SML macros and PC ARC/INFO includes an ARC.SML, ARCEDIT.SML and an ARCPLOT.SML in its directory structure.

ARCEXE\PTOOL\ARC\ARC.SML
ARCEXE\PTOOL\UTOOL\ARCEDIT.SML
ARCEXE\PTOOL\UTOOL\ARCPLOT.SML

There are two ways to launch module specific initialization files:

- 1) Rename your old SML file and enter it on the command line (i.e., ARCEDIT startup.fil).
If you have an application that uses a default ARC.SML, ARCEDIT.SML and ARCPLOT.SML, rename the SML and change the command lines in your macro(s) to include the new file names.
- 2) Create a special initialization file at your workspace directory that will automatically be run when the module command is entered.
If a file named ARC.INI, ARCEDIT.INI or ARCPLOT.INI is located in your 'start-up' directory, it will automatically be processed when the module name is started. PC ARC/INFO includes default initialization files that set the default dialog color and symbol sets for a PC ARC/INFO session. If a customized file is not located in the start-up directory, the default file gets run instead.

Here are the default initialization files and their contents. They establish the default settings when ARC, ARCEDIT and ARCPLOT are started:

```
ARCEXE\PTOOL\ARC\ARC.INI:      DIALOGCOLOR 1 4
                                &RET

ARCEXE\PTOOL\ARCEDIT\ARCEDIT.INI:  LINESET COLOR
                                MARKERSET COLOR
                                TEXTSET PLOTTER
                                &RET

ARCEXE\PTOOL\ARCPLOT\ARCPLOT.INI:  SHADESET COLOR255
                                LINESET PLOTTER
                                MARKERSET COLOR
                                TEXTSET PLOTTER
                                &RET
```

Changing workspace and drive locations with &WS, A and CD

1) For continuity with past versions of PC ARC/INFO, a version of the DOS CD command has been retained in version 4.0. Note, though, that, unlike the DOS version, a space must exist between the command name and the following parameter. This includes the backslash and dot options. Except for the {sml_var} option, the CD command has the same usage as the &WS directive.

CD {pathname}

Examples:

```
CD ..      move up one directory level
CD d:\     change to the D drive
```

2) A new SML directive has been added to navigate workspaces:

&WS {pathname} {sml_var}

pathname - the path or workspace to move to.

sml_var - an optional SML variable to assign the result to. If {pathname} does not exist, {sml_var} is assigned -1.

Entering &WS without a {pathname} or {sml_var}, will cause the current Workspace pathname to be echoed on the screen. You may use the backslash (\) convention to refer to the root director, the double dot (..) convention to refer to the parent directory and a single dot (.) to specify the current workspace path.

Examples:

```
&WS          displays current pathname on the screen
&WS . -10    prints current pathname into {sml_var} -10
&WS c:\mydir move to c:\mydir
&WS xxx -10  move down to xxx and store new pathname in {sml_var} -10
&WS ..      move up one directory level.
```

3) The A utility remains unchanged.

New directory name for temporary ARC files

Temporary ARC files are now written to a directory named T_ARCWn (where 'n' is the number of the current text window). This directory resides in the WKSP directory. Previously, the directory was named with a dollar sign (T\$ARCn) instead of the underscore character (_). This change allows PC ARC/INFO 3.x and 4.x to be running at the same time. It also allows temporary .DBF files to be written to the WKSP directory.

WKSP directory now includes a list of installed fonts

A new file named T\$FONTS.LST is also written to the WKSP directory when PC ARC/INFO starts up. This file contains a list of all the fonts installed on your computer. There is one font number and name per line. The font number is left justified in columns 1-4 and the font names are in columns 6-37. This file will automatically be deleted when you quit ARC.

On-line Help file has new organization

All ARC Internal and External commands are listed under the ARC Command Reference. This includes those previously listed as Overlay and Data Conversion commands. The previous distinction in functionality is no longer appropriate, especially since all ARC External commands are available to ARCEDIT and ARCPLOT. The ARCEDIT and ARCPLOT command references include their Internal commands only. Commands listed as External for ARCEDIT and ARCPLOT are described under the ARC Command Reference.

Commands that have been removed or replaced

ROUTE and ALLOCATE have been removed

These NETWORK programs have been removed from PC ARC/INFO:

ABUILD	ACREATE	ADDBACK	ADDFIX	AERROR
ALLOCATE	ALLOCATW	AMATCH	APARSE	ROUTE
ROUTEW	TURNTABLE	@WALLOC	@WROUTE	

Also removed from ARCPLOT: ADDRESSCOV ADDRESSES WHEREIS

This means that you will not be able to perform Network, address matching or geocoding functions with PC ARC/INFO 4.0. If a single precision coverage containing files used by these processes is converted to a double precision coverage, those files will be carried along "as is", but will not be usable in 4.0. If the coverage is exported or copied back to a single precision coverage, those files will again be available for use in PC ARC/INFO 3.5.x.

The following PC ARC/INFO commands have been removed or replaced

Previous versions of PC ARC/INFO included utilities to help manage ARC sessions in DOS. These are longer supported and are included in the list of ARC commands that have been removed. In addition, there are several new features of Version 4.0 that replace older functions. Commands that no longer exist but which have a direct replacement command in Version 4.0 are listed with the ARC commands that have been replaced. If you have existing applications that use any of the commands from the following lists, you will need to make the appropriate changes.

These ARC commands have been removed:

&GETXYC	Outdated DOS utility.
&HELP	Outdated DOS utility. Use F1 key for On-line HELP
&SETKEY (&SK)	Outdated DOS utility
&KEY {NW}	Outdated option to the &KEY directive
ADS	Outdated menu interface. Use ARCEDIT menu utilities.
ARC33-34	Outdated conversion utility
BMPDIS	Outdated DOS utility
CAL5800	Outdated DOS utility
CGITEST	Outdated DOS utility
CON-CGI	Outdated DOS utility
CONVSML	SML now supports Named Variables
DIGITIZE	Outdated menu interface. Use ARCEDIT menu utilities.
DRIVERS	Outdated DOS utility
DUALMODE	Outdated DOS utility
EDIT	Outdated menu interface. Use ARCEDIT utilities.
EDITPLOT	Outdated DOS utility. Use ARCPLOT utilities.
F	Outdated utility
FLST	Outdated utility
FM	Outdated utility
FONTEDIT	Outdated utility
HELP	Outdated DOS utility. (Also GENERATE HELP) Use the on-line HELP
HPGL2	Outdated DOS utility.
INDEXINFO	Use the new Windows command INDEXEDIT
INFODB	Outdated utility
KERMIT	Outdated utility
PACK	Outdated utility
PCX	Outdated format
PKUNZIP	Outdated utility
PKZIP	Outdated utility
PLOT	Outdated utility. Use your Windows Print utility.
PLOT33	Outdated utility
RENUMWID	Outdated utility
RUN	Outdated utility
SCREEN	Outdated utility
SCREENREST	Outdated format file
SCREENSAVE	Outdated format file
	<i>SCREENRESTORE and SCREENSAVE are only used in ARCEDIT and ARCPLOT. The ARC .RAS file format is outdated. Use the keyboard PrintScreen key to copy from the screen to the Windows clipboard. Alt-PrintScreen will copy everything within the active window to the clipboard. PrintScreen will copy the whole screen to the clipboard</i>
UNPACK	Outdated utility.

These ARC commands have been replaced

3.5.2 command: Replaced with this 4.0 command:

ARCEDITW	ARCEDIT
ARCPLOTW	ARCPLOT
COLUMNS	ITEMS {ALL / RELATE}
CON-DIG	CONDIG

DBASE	&SYS DBASE
DELETE	DEL, ERASE or DELTREE
DELETETIC	DELREC
DIGTESTW	DIGTEST
DRAW	PLOT (ARCPLOT) or SCREENRESTORE
DRAWW	PLOT (ARCPLOT) or SCREENRESTORE
G	CD or &WS
INDEXLIS	INDEXEDIT
JOIN	RELATE
LINEEDIT	SYMEDIT
LOG	&ECHO &COM or &ECHO &WATCH
LQ	WIN FILE or INFODIRECTORY or L
POPUP	WIN
SL	&DISPLAY
RESELECT	EXTRACT (uses current selection set)
RUN	&SYS or &
SIZEITEM	MODITEM
SYMINFO	SYMEDIT
TABLES	replaces the TABLESW command. TABLES subcommands no longer exist. All TABLES functions are now ARC level commands that can also be used in ARCEDIT and ARCPLOT. The next section describes this change.
TABLESW	TABLES
TCOPY	Use: & XCOPY
TREN	RENAME
W	CD or &WS
WAIT	WIN IDLE [seconds]
WIN SEL	READSELECT and WRITESELECT
@WEDIT	EDITMENU
@WPLOT	PLOTMENU
@WSHELL	ARCMENU
@PAT	SYMEDIT
@COLORS	SYMEDIT

These ARCPLOT commands have been removed

COLOR
PENSPEED
PLOTZOOM

These ARCEDIT and ARCPLOT commands have been replaced

COLUMNS	ITEMS {ALL / RELATE}
JOIN	RELATE
COLOR	SYMEDIT

These SML Developer Toolkit macros have been removed

ADDNPNTS	CD	COVFEAT	DISPTIME	DRAWBARS
GETNUM	GETRNG	GETSRT	GETSTRF	GETWORD

HISTOGRM	HMINMAX	ITMFUNIQ	MESSDLOG	STRAC2SP
STRSP2AC	SYSFM			

It should also be noted that the remaining SML Toolkit macros have been updated for version 4.0. The parameters and results of these routines may have changed slightly from previous versions.

These SML Developer Toolkit macros have been replaced

ISJOIN is now ISRELATE
PAUSE is now WAIT

TABLES subcommands have been replaced with ARC level commands

There is a new Windows version of TABLES that replaces the old TABLESW. In addition, the TABLES subcommands have been replaced with ARC level commands. Data file processing is now performed at the ARC, ARCEDIT and ARCPLOT prompts. These commands are universally available to all modules, although the syntax of some of them may be internal to a specific module. For instance, the ITEMS command does not require any arguments in ARCEDIT, because the only data file that can be listed is the attribute file of the current edit coverage. Using ITEMS at the ARC or ARCPLOT prompts does require command arguments, although even with these two modules, the information required to process the ITEMS command is slightly different.

Below is the list of TABLES commands and their Version 4.0 equivalents.

A	ARC command
AE	ARCEDIT command
AP	ARCPLOT command

Unchanged	- the command existed at 3.5.2 and was converted without any change
Modified	- the command existed at 3.5.2, but required modification
New	- the command is new for Version 4.0

<u>TABLES 3.5.2</u>	<u>ARC 4.0</u>	<u>Unchanged</u>	<u>Modified</u>	<u>New</u>
\$KILL	- KILLINFO	A		AE / AP
ADD	- ADDREC {DELIM option}			A / AE / AP
	- FORMS {ADD option}	AE	AP	A
ADDITEM	- ADDITEM		A	AE / AP
ASELECT	- ASELECT	AE	AP	A
CALCULATE	- CALCULATE	AE	AP	A
CLEAR	- CLEAR		A / AE / AP	
	- DIALOGCLEAR	AE / AP		A
COLOR	- COLOR		A	
	- DIALOGCOLOR		AE / AP	A
COLUMNS	- ITEMS {ALL RELATE}		A / AE / AP	
COMMANDS	- COMMANDS		A / AE / AP	
DEFINE	- DEFINE			A / AE / AP
DIRECTORY	- INFODIRECTORY	A		AE / AP
DROPITEM	- DROPITEM		A	AE / AP

DUMP	- PUTREC	A / AE / AP
ERASE	- KILLINFO A.....	AE / AP
FORMS	- FORMS AE..... AP.....	A
GET	- ADDRREC {INFO option} ...	A / AE / AP
HELP	- [dropped]	
ITEMS	- ITEMS	A / AE / AP
JOIN	- RELATE	A / AE / AP
LIST	- LIST AE / AP	A
LOAD	- ADDRREC {FIXED option} .	A / AE / AP
MOVEITEM	- MOVEITEM	A / AE / AP
NSELECT	- NSELECT..... AE / AP	A
PACK	- PURGEREC	
POPUP	- [dropped]	
PURGE	- DELREC.....	A / AE / AP
QUIT	- QUIT	A / AE / AP
RESELECT	- RESELECT AE..... AP.....	A
SAVE	- ADDRREC {INFO option} ...	A / AE / AP
	- DEFINE {with 2 nd option}..	A / AE / AP
SCREENRESTORE	- [dropped]	
SCREENSAVE	- [dropped]	
SELECT	- SELECT AE.....	A / AP
SORT	- SORTREC.....	A / AE / AP
STATISTICS	- STATISTICS..... AP.....	A
STOP	- [dropped]	
SYSTEM	- &	A / AE / AP
	- &SYSTEM.....	A / AE / AP
	- SYSTEM	A / AE / AP
UNSELECT	- UNSELECT AE.....	A / AP
UPDATE	- FORMS with record# AE..... AP.....	A
	- CALCULATE AE.....	A / AP
	- MOVEITEM AE.....	A / AP

For an overview of how these commands are implemented at the ARC level, refer to the section “An introduction to the ARC level database management commands” later in this guide.

Usage changes for existing commands

ARC usage changes

ADDITEM - dropped the [output_width] parameter

ADDITEM [in_file] [out_file] [item_name] [item_width] [item type] {decimal places} {start_item}

COLOR - support for foreground and background text colors
 COLOR [foreground_color] {background_color}

CONDIG - WinTab support added
 CONDIG [wintab_digitizer] - WinTab support added

COPYCOV - support for double precision coverages
 COPYCOV [from_cover] [to_cover] {DOUBLE / SINGLE} {xshift}{yshift} {DOS / ANSI}

DROPITEM - up to 10 items can be listed
 DROPITEM [in_file] [out_file] [item1] {item2...item10}

ELIMINATE - subcommand processing dropped - now uses the current selection set
 ELIMINATE [in_cover] [out_cover] {KEEPEdge}

EXPORT - removed the DOUBLE option
 EXPORT [COVER / INFO / TEXT / PLOT / MAP] [input] [interchange_file] {N / P / F}
 {max_lines}

EXTRACT - subcommand processing dropped - now uses the current selection set
 EXTRACT [in_cover] [out_cover] {POLY/LINE/POINT}

GENERATE - new subcommands
 COMMAND / CURVES / POLYGONS

IMPORT - removed the SHIFT option
 IMPORT [COVER / INFO / TEXT / PLOT / MAP] [interchange_file] [output]

INDEXITEM - writes the index file to a file named *info_file*.DBX
 INDEXITEM [info_file] [item]

INFODIRECTORY - support for formatted output and RELATE status
 INFODIRECTORY {path} {BRIEF / LIST / LONG} {ALL / ON / OFF}

ITEMS - dropped keyword NONE - support for related items
 ITEMS [info_file] {ALL/ITEMS/LIST/TYPE} {ALL/NORELATE/RELATE}

L - many new LIST options including output to a file

LIST - dropped keyword NONE
 LIST [info_file] {range} {item...item}

LISTCOVS - support for single and double precision coverages
 LISTCOVS [workspace] {DOUBLE / SINGLE}

LOADANNO - support for annotation subclasses
 LOADANNO [cover] {subclass}

MIFSHAPE - directs MIF file information to a file
 MIFSHAPE [INFO] [mif_file] {output}

MODITEM - command line processing - no subcommands
 MODITEM R [info_file] [item] [new_width] {type} {decimals}
 MODITEM M [info_file] [item] {new_item} {type} {decimals}
 MODITEM S [info_file] [item] [new_item1] [width] [new_item2]
 MODITEM C [info_file] [item1] [item2] [new_item]

UNGEN - support for polygon features and coordinate representation
 UNGEN [LINE / POINT / POLY / TIC] [cover] [file] {NODES / NONODES}
 {EXPONENTIAL / FIXED}

UNLOADANNO - support for annotation subclasses

UNLOADANNO [cover] {subclass} {#_points}

WIN - new options

WIN COLOR [* / [R / N] [file]] - the * option displays colors 1 to 255 in a dialog box. The selected color is returned to variable 1.

WIN EXEC 9 {P} - returns a 5 digit font number (ffff). If the {P} parameter is used, an 8 digit number is returned (pppffff), the first 3 digits representing the pattern number (32 - 255) of the selected symbol from that font.

WIN FILE T [*] [title] - Allows the user to pick a coverage having ANNOTATION

WIN IDLE {seconds}

&KEY - removed &NW option

&KEY [var] {string}

ARCEDIT usage changes

BACKENVIRONMENT - image support

BACKENVIRONMENT [{ALL {ON / OFF / IDS}} {ARC {ON / OFF / IDS / ARROWS}} {NODE {ON / OFF / ERRORS / DANGLE / PSEUDO / ALL}} {LABEL {ON / OFF / IDS}} {TIC {ON / OFF / IDS}} {ANNO {ON / OFF / level1... level20}} IMAGE {ON / OFF}}]

DIALOGCOLOR - - support for foreground and background text colors

DIALOGCOLOR [foreground_color] {background_color}

DISPLAY - dropped DOS supported features.

DISPLAY [4] {option} {dialog_lines}

DISPLAY [0]

EDITFEATURE - supports annotation subclasses

EDITFEATURE [NONE/ARC/NODE/LABEL/TIC/ANNO] {subclass}{Y}]

GET - support for annotation subclasses

GET [cover] {subclass}

ITEMS - lists items on any data file as well as the current edit coverage. Supports related items.

ITEMS {info_file} {ALL/ITEMS/LIST/TYPE} {ALL/NORELATE/RELATE}

MAPEXTENT - image and annotation subclass support

MAPEXTENT [* / DEFAULT / ZOOM {scale_factor}]

MAPEXTENT {BND / TIC} [cover...cover]

MAPEXTENT IMAGE [image...image]

MAPEXTENT [xmin ymin xmax ymax]

MOVEITEM - support for text expressions

MOVEITEM [text_exp] {TO} [item / sml_var] {str_ops}

PUT - support for annotation subclasses

PUT [cover] {subclass}

SCREENRESTORE - restores the SCREENSAVE display list file

SCREENRESTORE [filename]

SCREENSAVE - saves the entire graphic screen

SCREENSAVE [filename]

SHOW - support for new commands

ANNO n ALIGNMENT {1} - annotation alignment of the nth annotation

ANNO [n] ID - User-ID of the nth annotation

ANNO [n] JUSTIFY - annotation positioning of the nth annotation
 ANNOALIGNMENT {1} - current annotation alignment
 BACKENVIRONMENT {6} - background environment for arcs, nodes, labels, tics, annotation and images.
 ANNOSUBCLASS [0-4] {1} - name of the current annotation subclass for an edit coverage
 DIALOGCOLOR {2} - foreground and background colors of text in the dialog window.
 IMAGE [1-4] {1} - name(s) of the image(s) in the background image list.
 LABEL [n] ANGLE - angle of the label ID of the nth label.
 LABEL [n] SCALE - scale factor of the label ID of the nth label.
 LINEINFO [symbol] [layer] {8} - characteristics of a line symbol
 LINESET {1} - name of the current lineset file.
 MARKERINFO [symbol] {4} - characteristics of a marker symbol
 MARKERSET {1} - name of the current markerset file.
 TEXTINFO [symbol] {9} - characteristics of a text symbol
 TEXTSET {1} - name of the current textset file

ARCPLOT usage changes

ANNOCOVERAGE - subclass support
 ANNOCOVERAGE [cover {subclass} / NONE]
 ANNOTEXT - subclass support
 ANNOTEXT [cover] {subclass / ALL} {level...level}
 ARCTEXT - support for concatenated and stacked text
 ARCTEXT [cover] [text_expression] {lookup_table} {POINT1 / POINT2 / LINE}
 {UR/UC/UL/CR/CC/CL/LR/LC/LL/BLANK} {NOFLIP}
 ASELECT - support for POINT, OVERLAP and feature on feature selection
 ASELECT [cover / info] [feature_class] {logical_expression}
 ASELECT [cover] [feature_class] POINT [xy / *] {PASSTHRU / WITHIN}
 ASELECT [cover] [feature_class] BOX [xmin ymin xmax ymax / *] {PASSTHRU / WITHIN}
 ASELECT [cover] [feature_class] POLY [x y...x y / *] {PASSTHRU / WITHIN}
 ASELECT [cover] [POLYS] ADJACENT {distance / *}
 ASELECT [cover] [feature_class] CIRCLE [x y radius / * {radius}] {PASSTHRU / WITHIN}
 ASELECT [cover] [feature_class] MAPEXTENT
 ASELECT [cover] [feature_class] OVERLAP [ovr_cover] [ovr_feature_class] {PASSTHRU / WITHIN}
 CALCULATE - will operate on a data file as well as a coverage attribute table
 CALCULATE [cover] [feature_class] [target_item] [=] [expression]
 CALCULATE [cover] [feature_class] [sml_var] [=] [expression]
 CALCULATE [info_file] [NONE] [target_item] [=] [expression]
 CALCULATE [info_file] [NONE] [sml_var] [=] [expression]
 CIRCLE - will draw a circle, an arc or a 'pie' segment
 CIRCLE [x y radius / * {radius}] {start_angle} {end_angle} {ARC / LINE / SHADE}
 CLEARSELECT - will optionally clear only one coverage or file
 CLEARSELECT {cover {feature_class} / info_file}
 DIALOGCOLOR - - support for foreground and background text colors
 DIALOGCOLOR [foreground_color] {background_color}
 DISPLAY - dropped DOS supported features.

DISPLAY [4] {option} {dialog_lines}
 DISPLAY [0]

DROPLINE - support for stacked and concatenated text
 DROPLINE [cover] [text_expression] {lookup_table} {NOTEXT}

FORMS - new option to ADD records
 FORMS [info_file] [NONE] [ADD / record] {item...item}

ITEMS - lists items on any data file as well as a coverage. Supports related items.
 ITEMS [cover] [feature_class] {ALL/ITEMS/LIST/TYPE}{ALL/NORELATE/RELATE}
 ITEMS [info_file] {INFO} {ALL/ITEMS/LIST/TYPE} {ALL/NORELATE/RELATE}

ITEMTEXT - support for concatenated and stacked text
 ITEMTEXT [info_file] [text_expression] [record] [*/xy]

KEYBOX - can use the screen cursor to define the size of the box
 KEYBOX [* / width height]

KEYSEPARATION - can use the screen cursor to define the distances
 KEYSEPARATION [* / horizontal vertical]

LABELTEXT - support for concatenated and stacked text
 LABELTEXT [cover] [text_expression] {lookup_table}

LINESET - will load specific symbols from a symbol file to the current line set
 LINESET [* / [lineset_file] {start_sym} {end_sym} {new_start#} {CLEAR}]

LINESIZE - can use the screen cursor to define the size
 LINESIZE [width / *]

MAP - opens a map without clearing the graphic screen
 MAP [map_name / END] {DRAW / NODRAW / NOCLEAR}

MAPEXTENT - image and annotation subclass support
 MAPEXTENT {feature_class} [cover...cover]
 MAPEXTENT ANNO [cover{.subclass}...cover{.subclass}]
 MAPEXTENT [xmin ymin xmax ymax / *]
 MAPEXTENT IMAGE [image...image]

MARKERFONT - can now specify a font by its name:
 MARKERFONT [font# / * / fontname {fontstyle}]

MARKERSET - will load specific symbols from a symbol file to the current marker set
 MARKERSET [* / [markerset_file] {start_sym} {end_sym} {new_start#} {CLEAR}]

MARKERSIZE - can use the screen cursor to define the size
 MARKERSIZE [size / *]

MOVEITEM - support for stacked text and string parsing. Any data file can be used.
 MOVEITEM [cover] [feature_class] [text_exp] [TO] [item / sml_var] {str_ops}
 MOVEITEM [info_file] [NONE] [text_exp] [TO] [item / sml_var] {str_ops}

OVERAREA - can use the screen cursor to define the size of the area
 OVERAREA [* /width height]

OVERPOSITION - will set the overposition to the previous setting
 OVERPOSITION [* / RESET / x y]

OVERSEPARATION - can use the screen cursor to define the distances
 OVERSEPARATION [* / horizontal vertical]

POINTTEXT - support for stacked and concatenated text
 POINTTEXT [cover] [text_expression] {lookup_table}

POLYGONTEXT - support for stacked and concatenated text
 POLYGONTEXT [cover] [text_expression] {lookup_table}

RESELECT ARC PLOT - support for POINT selection and feature on feature selection
 RESELECT [cover / info] [feature_class] {logical_expression}
 RESELECT [cover] [feature_class] POINT [xy / *] {PASSTHRU / WITHIN}
 RESELECT [cover] [feature_class] BOX [xmin ymin xmax ymax / *] {PASSTHRU / WITHIN}
 RESELECT [cover] [feature_class] CIRCLE [x y radius / * {radius}] {PASSTHRU / WITHIN}
 RESELECT [cover] [feature_class] POLY [x y...x y / *] {PASSTHRU / WITHIN}
 RESELECT [cover] [feature_class] MAPEXTENT
 RESELECT [cover] [feature_class] OVERLAP [ovr_cover] [ovr_feature_class] {PASSTHRU / WITHIN}

SCREENRESTORE - restores a display list file to the graphic screen
 SCREENRESTORE [filename]

SCREENSAVE - saves the whole graphic screen to a display list file
 SCREENSAVE [filename]

SHADESET - will load specific symbols from a symbol file to the current shade set
 SHADESET [* / [shadeset_file] {start_sym} {end_sym} {new_start#} {CLEAR}]

SHOW - support for new commands
 DIALOGCOLOR (2) the foreground and background colors
 ITEMS [cover / data_file] [feature] [item#] (4) - definition of an item
 LINEINFO [sym] [layer] (8) characteristics of a line symbol
 LINESET (1) - the name of the current line set
 MARKERINFO [sym] (4) - characteristics of a marker symbol
 MARKERSET (1) - the name of the current marker set
 OVERBOX (4) - minimum and maximum coordinates of the overflow label column.
 SELECT [cover / data_file] [feature] (2) - number of selected and total records of a file
 SHADEINFO [symbol] (17) - characteristics of a shade symbol
 SHADESET (1) - the name of the current shade set.
 TEXTINFO [symbol] (9) - characteristics of a text symbol
 TEXTSET (1) - the name of the current text set

TEXTFONT - can now specify a font by its name:
 TEXTFONT [font# / * / fontname {fontstyle}]

TEXTINDEX - will also display symbols from the current textset
 TEXTINDEX [xmin ymin xmax ymax / *] {ALL}
 TEXTINDEX [xmin ymin xmax ymax / *] [begin] {end} {columns} {text}

TEXTSIZE - can use the screen cursor to define the size
 TEXTSIZE [* / [[height] {width}]]

New Commands

New Global commands

ADDREC - adds records from a data file or a text file to a database data file.

CD - shortened name for CHDIR

CHDIR - changes your current directory.

CLASSIFY - displays a dialog window in which numeric classifications can be modeled, viewed and saved.

COPYF - copies one or more files.

COPYWKSP - copies or converts all coverages on a workspace directory to another workspace directory in either single or double precision.

CLS - clears the command text window

DEL - deletes the specified file.

DEFINE - creates a new database data file template.

DELREC - deletes the currently selected records of a database data file.

DELTREE - deletes the specified directories.

DIR - displays a directory listing.

ERASE - deletes a disk file.

INDEXEDIT - interactively creates and removes indexed items.

ITEMEDIT - interactively edits item definitions.

KILLANNO - deletes the annotation of a coverage.

KILLINDEX - removes an attribute index from a data file or a spatial index from a coverage.

LISTANNO - displays the annotation subclasses of a coverage.

MD - shortened name for MKDIR

MENU - displays the ARC, ARCEDIT or ARCPLOT menu.

MKDIR - creates a new workspace directory.

PAUSE - causes the ARC, ARCEDIT or ARCPLOT processor to temporarily halt.

PUTREC - writes the selected records and items from a data file to a text file.

PURGEREC - permanently removes data file records which have been marked for deletion.

RD - deletes an empty directory.

RELATE - establishes, manages and removes permanent multilevel relates.

REEDIT - displays an interactive dialog box for the RELATE command.

REN - renames a disk file or directory.

RENAME - renames a disk file or directory.

RENANNO - renames an annotation subclass.

RM - shortened name for RMDIR

RMDIR - deletes an empty directory.

SHAPEDXF - displays information about an ArcView shapefile or converts a shapefile into a DXF file.

SORTREC - sorts the selected records of a data file.

TEXTANSI - converts the text in a text file, data file or PC ARC/INFO 3.x annotation file from DOS to ANSI.

TEXTDOS - converts the text in a text file, data file or PC ARC/INFO 3.x annotation file from ANSI to DOS.

TEXTTYPE - determines whether the format of the text in a text file, data file or PC ARC/INFO 3.x annotation file is ANSI or DOS.

TIGERTOOL - converts coverages created by TIGERARC into one polygon/line coverage (and an optional point coverage) containing all associated line and area attributes.

UNLOCK - unlocks read-only coverages and coverage files.

New ARC commands

ARCMENU - displays the menu interface for ARC.

ASELECT - adds records to the currently selected set of records.

CALCULATE - calculates and assigns values to an item in a database table or an SML variable.

CLEARSELECT - clears all selection sets from the current workspace, or the selection set of a specified database file.

DIALOGCLEAR - clears the text window.

DIALOGCOLOR - specifies the text and background colors of the text area.

FORMS - interactive modification of a data file record using an input form.

LISTSELECT - lists all current selection sets.

MENU - displays the ARC, ARCEDIT or ARCPLOT menu.

MOVEITEM - moves the contents of a text string to an item or an SML variable.

NSELECT - switches the selected and unselected sets of records in a database file.

READSELECT -

RESELECT - restores a selection set file.

SELECT - creates a selected set of records from a database data file.

SHOW - returns information about an ARC session.

STATISTICS - generates summary statistics for an item in the selected database file.

UNSELECT - removes records from the currently selected set of records for a data file.

WRITESELECT - writes the currently selected set of features to a selection set file.

New GENERATE subcommands

COMMANDS - lists the names of the available GENERATE subcommands.

CURVES - generates curves using the specified grain value as the distance between vertices on each curve.

POLYGONS - adds polygons and label points to the coverage.

New ARCEDIT commands

ANNOALIGNMENT - establishes how blocks of text or stacked annotation will be drawn.

AP - starts an ARCPLOT session without altering the current ARCEDIT screen environment and writes the session to the ARCEDIT DRAW environment.

ARCPLOT - starts an ARCPLOT session without altering the current ARCEDIT screen environment.

EC - shortened name for EDITCOVERAGE

EDITMENU - displays the menu interface for ARCEDIT.

IMAGE - adds an image file to the background display list.

LINESAVE - saves the current LINESET symbol table to a data file in ARCEDIT.

MARKERSAVE - saves the current MARKERSET symbol table to a data file.

READSELECT - restores a selection set file.

REMOVEIMAGE - removes an image file from the background display list.

SAVEFEATURE - saves the current edit feature of the current edit coverage.

TEXTINDEX - displays characters from ANNOSYMBOL 1 or text symbols of the current TEXTSET.

TEXTSAVE - saves the current TEXTSET symbol table to a data file.

WRITESELECT - writes the currently selected set of features to a selection set file in ARCEDIT.

New ARCPLOT commands

IMAGE - draws an image BMP file.

LINESAVE - saves the current LINESET symbol table to a data file.

LISTSELECT - lists all current selection sets.

MARKERSAVE - saves the current MARKERSET symbol table to a data file.

OVERBOX - sets the position and size of the label overflow column.

PLOTMENU - displays the ARCPLOT menu.

READSELECT - restores a selection set file.

SCALEBAR - draws a scalebar in ARCPLOT.

SELECT - creates a selected set of coverage features or data file records.

SHADESAVE - saves the current SHADESET symbol table to a data file in ARCPLOT.

TEXTALIGNMENT - establishes how blocks of text or stacked annotation will be drawn.

TEXTSAVE - saves the current TEXTSET symbol table to a data file in ARCPLOT.

UNSELECT - removes selected features from the currently selected set of features in ARCPLOT.

WRITESELECT - writes the currently selected set of features to a selection set file in ARCPLOT.

New commands for ARCEDIT and ARCPLOT

SYMEDIT - displays an interactive symbol editor.

SYMPICK - displays an interactive symbol picker.

New SML commands

&ARCPROG - runs internal programs or programs on the PROGRAMS subdirectory.

&IF - block IF statement.

&VAR - define or delete a named variable.

&WIN - provides a suite of functions for interfacing with Microsoft Windows.

&WS - move to a different workspace or save a workspace location.

New SML Developer's Toolkit commands

MAPENV - saves the page environment with a map composition.

STRCAP - capitalize the first letter of each word.

STRLOW - converts a string to lower case.

PC ARC/INFO 4.0 Limits

New:

Increased number of ARCS per POLY from 10,000 to 50,000.

Increased points per polygon for hardware shading from 8,191 to 16,125.

Increased max columns of grid files from 5,000 to 10,000.

Increased number of concurrent related files from 1 to 9.

Increased SML local variables from 20 to 50.

Increased SML maximum calling levels from 9 to 12.

Increased width of SML variables from 80 to 256 characters.

Increased command line length from 256 to 1024 characters.

Increased path name length from 64 to 256 characters.

Increased number of annotation position points saved by UNLOADANNO from 4 to 85

Increased number of points for ATLASARC to 32,761 points per feature.

Increases number of points for MIFSHAPE from 32,763 to 200,000.

Increased nodes in ARCDLG and ARCDLGN from 32,000 to 100,000.

Increased number of Images in ARCEDIT from 0 to 4.

Increased number of annotation position points from 4 to 500

Increased number of annotation arrow points from 3 to 5

Increased number of symbols in a LINESET file.

You can have up to 255 symbols, each with three layers.

Increased number of characters WIN CB can handle from 80 to 256.

Increased number of characters for CTRL-V paste (clipboard paste) from 80 to 1024.

Increased maximum number of rows in a dialog box from 25 to 29.

Increased maximum number of dialog groups from 6 to 10.

Increased range of scroll bars from (-32767 to 32767) to (-2**31 to 2**31).

Increased maximum number of internal/external commands from 300 to 1024.

Increased length of text that can be displayed with TEXT and TEXTFILE from 120 to 256.

Long path support. Pathnames can be up to 256 characters. Previously it was 64.

Increased VFILE handles to 15.

Increased MFILE handles to 30.

Increased ISP handles to 10.

Decreased EDGEMATCH arcs in the 'edge box' from 256K to 64K.

Unchanged:

- Maximum of 10 concurrent ARC sessions.
- Maximum of 256K features in a coverage.
- Maximum of 255 symbols in a symbol table.
- Maximum of 16 non-Update Zoom Windows
- Up to 3 concurrent SML &OPEN files.
- Up to 3 concurrent SML &OPENW files at once.
- Up to 10,000 global SML variables (0-9999).
- Up to 4 EDITCOVERAGES and 4 BACKCOVERAGES in ARCEDIT.
- Up to 255 widgets on one dialog box.
- Up to 255 choices on a menu.
- Up to 15 simultaneous dialog boxes in a group.
- Up to 949 simultaneous widgets in a group of dialog boxes.

Converting PC ARC/INFO Single Precision coverages to Double Precision

Conversion Issues

PC ARC/INFO 4.0 stores coordinate data with double precision accuracy (14-15 digits) and character data with the ANSI or ISO character set. Previous versions of PC ARC/INFO stored data in single precision (7-8 digits) with the OEM character set. Although there are a few commands that will work with both single and double precision coverages, PC ARC/INFO coverages will have to be converted to double precision before they can be used.

These commands will work with both Single and Double Precision Coverages:

L	LISTCOVS	COPYCOV	COPYWKSP	KILL
RENAMCOV	WIN FILE			

... and all commands that manipulate DBF files (i.e., ADDITEM, TABLES). *Note that some ANSI/OEM character set issues may apply to DBF file data. These issues are discussed below.*

Commands that Convert between Single Precision and Double Precision Coverages:

COPYCOV	COPYWKSP	IMPORT	EXPORT
---------	----------	--------	--------

Coverages can be Exported to an E00 file in one version and Imported as a coverage in another version, but this is the slowest method of converting data. It is much faster to use COPYCOV and COPYWKSP.

Using COPYCOV and COPYWKSP

COPYCOV and COPYWKSP are powerful tools to help you convert your single precision coverages to double precision. They will copy or convert one coverage (COPYCOV) or all coverages on a workspace directory (COPYWKSP) to another coverage or workspace directory in either single or double precision.

```
COPYCOV [from_cover] [to_cover] {DOUBLE / SINGLE} {xshift}{yshift} {DOS / ANSI}
COPYWKSP [from_wksp] [to_wksp] {DOUBLE / SINGLE} {xshift} {yshift} {DOS/ANSI}
```

[from_xxx] - the coverage or workspace to be copied

[to_xxx] - the resulting coverage or workspace after copying or converting

{DOUBLE / SINGLE} - the precision of the coverage(s) to be created. *Note that you can use the number 2 to denote DOUBLE precision or the number 1 for SINGLE precision wherever these keywords are used.*

{xshift} {yshift} - optional shift factor to be applied to coverage coordinates during precision conversion. If specified, they will be written to xyshift.dbf for each coverage.

{DOS/ANSI} - Specifies which character set to use for the single precision coverage when converting to or from single precision coverages.

DOS - the single precision coverage uses, or is to use, the DOS or OEM character set.

ANSI - the single precision coverage uses, or is to use, the Windows or ANSI character set.

Converting a coverage from single precision to double precision does not automatically increase the precision of the existing data. Data added to a single precision coverage remains accurate to single precision (7-8 digits) even after it is converted to double precision. Only the data added to a double precision coverage will be accurate to double precision (14-15 digits). Precision can be lost when converting a coverage from double precision to single precision.

Coordinate Shift Issues

COPYCOV and COPYWKSP will copy or convert coverages depending on the precision of [from_cover] and the option {DOUBLE / SINGLE}.

<u>from_cover</u>	<u>option</u>	<u>to_cover</u>	<u>operation</u>	<u>optional shifting</u>
single	single	single	copy	None
double	double	double	copy	None
single	double	double	conversion	Up (Add)
double	single	single	conversion	Down (Subtract)

Coordinate shifting only occurs during conversion. It does not occur for a copy. Shift values are added when a single to double conversion occurs. Shift values are subtracted when a double to single conversion occurs. Some single precision coverages may already have shift values stored in a file named XYSHIFT.DBF. This file is maintained in the coverage directory. If non-zero shift values are specified on the command line, they will be used even if a XYSHIFT.DBF file exists. If no shift values are specified (or they are 0,0), and there is a XYSHIFT.DBF file for the from_cover, then those shift values will be used. If shift values are used, they will be written to XYSHIFT.DBF for the to_cover.

OEM / ANSI / ISO International Character Issues

This issue is of most interest to those who use extended characters for display or data entry. Extended characters include special symbols and characters not in the English alphabet.

Double precision coverages store data using the ANSI character set and annotation using the ISO character set. Single precision coverages usually store this text using the OEM (DOS) character set. By default, converting from single to double precision coverages will perform an OEM to ANSI (or ISO) conversion on DBF and annotation files. Converting from double to single precision coverages will perform an ANSI (or ISO) to OEM conversion on DBF and annotation files.

This default behavior can be changed, if required, by using the {DOS/ANSI} option with the COPYCOV and COPYWKSP commands.

The ANSI option indicates that the single precision coverage is in ANSI (or you want it written to the single precision coverage as ANSI) and no conversion is done. This table shows the combinations.

<u>from cover</u>	<u>to cover</u>	<u>option</u>	<u>text conversion</u>
single	double	DOS	OEM to ANSI/ISO
double	single	DOS	ANSI/ISO to OEM
single	double	ANSI	None (both are ANSI/ISO)
double	single	ANSI	None (both are ANSI/ISO)

Warning: Converting a file with an ANSI character set to an ANSI character set or converting a file with an OEM character set to an OEM character set may cause problems with the extended characters. Use the TEXTTYPE command to determine the format of the text or annotation in a file. If required, use the TEXTDOS or TEXTANSI command to convert the text format of the file.

The current language of the operating system determines the ANSI and OEM code pages that will be used for the conversion. This can be changed by changing the language or changing the registry using the Windows command REGEDIT

There is a new subdirectory called LOCALE under the ARCEXE directory. This directory holds the lookup tables for ANSI to ISO and ISO to ANSI code pages used for annotation conversion. Annotation is stored with ISO code pages, but is used internally with ANSI code pages. In general, you should not edit these files, but they are included with PC ARC/INFO in the rare case that the ANSI/ISO conversion is not quite correct. The files and their meaning are listed in the following table:

FILE	FROM CODE PAGE	TO CODE PAGE	LANGUAGE
F1250	ANSI 1250	ISO 8859-2	Eastern European
T1250	ISO 8859-2	ANSI 1250	Eastern European
F1251	ANSI 1251	ISO 8859-5	Russian
T1251	ISO 8859-5	ANSI 1251	Russian
F1253	ANSI 1253	ISO 8859-7	Greek
T1253	ISO 8859-7	ANSI 1253	Greek
F1254	ANSI 1254	ISO 8859-3	Turkish
T1254	ISO 8859-3	ANSI 1254	Turkish
F1255	ANSI 1255	ISO 8859-8	Hebrew
T1255	ISO 8859-8	ANSI 1255	Hebrew
F1256	ANSI 1256	ISO 8859-6	Arabic
T1256	ISO 8859-6	ANSI 1256	Arabic
F1257	ANSI 1257	ISO 8859-4	Lithuanian
T1257	ISO 8859-4	ANSI 1257	Lithuanian

Each table consists of 128 numbers, one per line. These are character codes 128-255 of the FROM code page and the values represent the TO code page conversion value.

There are no F1252 and T1252 tables (ANSI 1252, ISO 8859-1, U.S.A and Western Europe) as these code pages are identical and no conversion is needed.

Converting TEXT and DATABASE files used by PC ARC/INFO

There are three utility commands for converting ANSI/ISO and OEM format files that PC ARC/INFO does not automatically convert when copying between single and double precision coverages. (See the OEM / ANSI / ISO discussion above.) One converts files to ANSI/ISO format. The other converts them to DOS (OEM) format and the third command helps identify the current format of the file.

```
TEXTANSI [TEXT/INFO] [in_file] {out_file}
TEXTANSI [ANNO] [cover]
```

The TEXTANSI command converts a text [TEXT] file, database file [INFO] or PC ARC/INFO 3.x annotation file [ANNO] from DOS (OEM) to ANSI.

```
TEXTDOS [TEXT/INFO] [in_file] {out_file}
TEXTDOS [ANNO] [cover]
```

The TEXTDOS command converts a text [TEXT] file, database file [INFO] or PC ARC/INFO 3.x annotation file [ANNO] from ANSI to DOS (OEM).

```
TEXTTYPE [TEXT / INFO] [in_file] {out_file}
TEXTTYPE [ANNO] [cover] {out_file}
```

The TEXTTYPE command will look at the text on a text file, INFO file or PC ARC/INFO 3.x annotation file and try to determine whether the text is ANSI or DOS (OEM). PC ARC/INFO 4.0 uses ANSI text while earlier versions of PC ARC/INFO generally used DOS.

Some coverage file names have changed

In converting single precision 3.x coverages to double precision 4.0 coverages, some coverage file names will be changed:

<u>Single</u>	<u>Double</u>	<u>Description</u>
AAT.DBF	AAT.DBF	Arc attributes.
AAT.INF	AAT.INF	AAT dBase/INFO conversion information
AAT.NDX	AAT.DBX	AAT item indices
	AAT.REL	New - AAT 'relate' information
	AAT.SEL	New - AAT 'selection set' information
ABN	ABN.ADF	Arc feature index.
ABX	ABX.ADF	Index for ABN.
ARC	ARC.ADF	Arc features.
ARX	ARX.ADF	Index for ARC.
BND.DBF	DBLBND.DBF	Boundary file.
CNT	CNT.ADF	Centroid features.
CNX	CNX.ADF	Index for CNT.
LAB	LAB.ADF	Label/Point features.
MSK	MSK.ADF	Partial processing mask.
PAL	PAL.ADF	Polygon topology file.
PAX	PAX.ADF	Index for PAL.
PAT.DBF	PAT.DBF	Point/Label attributes.
PAT.INF	PAT.INF	PAT dBase/INFO conversion information
PAT.NDX	PAT.DBX	PAT item indices
	PAT.REL	New - PAT 'relate' information
	PAT.SEL	New - PAT 'selection set' information
PBN	PBN.ADF	Polygon feature index.

PBX	PBX.ADF	Index for PBN.
PFF	PFF.ADF	Partial processing filter file.
PFX	PFX.ADF	Index for PFX.
PRJ	PRJ.ADF	Projection information
TOL	PAR.ADF	Tolerances file.
TBN	TBN.ADF	Annotation feature index.
TBX	TBX.ADF	Index for TBN.
TXT	TXT.ADF	Annotation features.
TXX	TXX.ADF	Index for TXX.
TIC.DBF	DBLTIC.DBF	Tic file.
XBN	XBN.ADF	Point feature index.
XBX	XBX.ADF	Index for XBX.

Some default item definitions have changed

Some default Numeric items in coverage data files will have their item width increased from N13 to N20 when the coverage is converted from single to double precision. The items in these files have new definitions:

<u>Single</u>	<u>Double</u>	<u>Items with new width of N20</u>
TIC	DBLTIC	XTIC, YTIC
BND	DBLBND	XMIN, YMIN, XMAX, YMAX
AAT	AAT	LENGTH
PAT	PAT	AREA, PERIMETER

What happens to Address and Geocoding files when a coverage is converted to PC ARC/INFO 4.0?

Network, address matching and geocoding functions are not supported with PC ARC/INFO 4.0. If a coverage containing files used by these processes is converted to a PC ARC/INFO 4.0 coverage, those files will be carried along "as is", but will not be usable in 4.0. If the coverage is copied back to its originating source, those files will again be available for use. These files include:

ADI	ADP	ADX	ADD.DBF	TRN.DBF
-----	-----	-----	---------	---------

What happens to single precision annotation now that PC ARC/INFO 4.0 supports annotation subclasses?

There will be no loss of information when converting annotation from single to double precision. All the annotation will be placed in the default or unnamed (#) annotation subclass. Annotation subclasses are described later in this document.

What happens to PC ARC/INFO 4.0 annotation when it is converted to single precision?

Double precision annotation supports many new annotations features that are not supported in single precision. If you add double precision annotation using any of these new features, there may be some loss of information, when you convert from double precision to single precision. Here is what happens to the new features on conversion from double to single. (These features are described later in this document):

Text alignment will default to LEFT.

Text position will default to LL.
Stacked annotation will display the character "\" instead of inserting a carriage return.
Text with more than 4 points are splined to 4 points.
Text arrows with more than 3 points are splined to 3 points.
User-IDs will be lost.
Text with multi-word positioning will be positioned at the first word.
All subclasses are merged into the one TXT file.

By default, all annotation files will be converted from the ISO character set to the OEM character set on conversion from double precision to single precision. Use the {DOS / ANSI} option on COPYCOV or COPYWKSP to change this default behavior. See OEM / ANSI / ISO Issues for more information.

A note about annotation that is brought in to PC ARC/INFO 4.0 with User Position and User Offset characteristics.

PC ARC/INFO does not support these annotation characteristics. However, this information is maintained with a PC ARC/INFO 4.0 coverage. If the coverage is converted back to its original source, these annotation characteristics can be restored. If a PC ARC/INFO 4.0 coverage that is maintaining this information and is converted to single precision, these rules follow:

Text with a user position will be flipped, if needed.
Text with a user position will lose the user position.
Text with a user offset will lose the offset.

New Windows TABLES command

Typing TABLES at any command prompt will display the new TABLES menu. It allows you to interactively manage the data and the items of a data file. It also includes display options for viewing part or all of your files based on record or item selection.

Managing data files with the new ARC level database management commands

The following is not a complete list, or a complete description, of the PC ARC/INFO database management commands, but it should help illustrate how the previous TABLES subcommands are implemented at the ARC level. Use the table included in the section "TABLES subcommands have been replaced with ARC level commands" earlier in this guide along with the on-line command references for full details. You will note from that table that data management commands are available in ARCEDIT and ARCPLOT as well.

Managing Data files

DEFINE [new_info_file] {old_info_file}

This command creates an empty database file. It is very similar to the TABLES DEFINE commands. The database file it creates does not contain any records. It will not contain any items, either, unless an {old_info_file} is specified. If an {old_info_file} is entered, its items and relations (established with the RELATE command) will also be copied to the new file. ADDITEM can then be used to add records to the new database file and ADDREC can add records.

Manipulating Records

ADDREC [to_file] [from_file] [INFO / FIXED]

[to_file] [from_file] [DELIM] {items...items}

ADDREC adds records from a data file or a text file to an existing database data file. The INFO option replaces the TABLES GET and SAVE commands and adds data from a data file. [FIXED] adds records from a fixed length text file (replacing the TABLES LOAD command) and DELIM works like the ADD FROM command and adds records from a delimited, or long, text file.

DELREC [info_file] {Y}

DELREC deletes the selected records from the database file [data_file]. It will prompt you for confirmation before deleting any records. You can enter Y (Yes) on the command line to suppress this prompt. On completion of the command, all remaining records become selected.

FORMS [info_file] {ADD / record} {items...items}

This is a new command for [ARC] and has been updated for[ARCPLOT]. It displays an input form on the screen that lets you modify the data of an existing record or add a new record to a data file.

PURGEREC [info_file] {Y}

PURGEREC permanently removes records which have been marked for deletion from a database file. In dBASE and some other data management packages, deleting records from a database file is a two-step process. First, records are marked for deletion and then another command must be executed to actually remove the records from the file. If this second step is not performed, the marked records physically remain in the file and can be erroneously accessed by PC ARC/INFO commands.

PUTREC [from_file] [to_file] [DELIM / FIXED / LONG / ITEM] {items...items}

This command replaces the DUMP program in TABLES. It writes out selected records and items from a data file to a text file in four different formats. Related items, concatenated items and range items can all be used. If no items are specified, then all items (including related items) will be written.

SORTREC [data_file] [item1] {A / D} {item2 {A / D} ... item10 {A / D}}

Sorts the selected records of a data file by the specified items. You may have up to 10 items. Use "A" for ascending sort, "D" for descending sort. An Ascending sort is the default.

Manipulating Items

ADDITEM, DROPITEM and MODITEM are all similar to previous versions of the commands, but, as noted earlier, their usages have been updated for version 4.0.

ITEMEDIT is a new command. It opens a dialog box on the screen that displays all the items for a data file. You can then interactively add, modify, delete, combine and split items.

There is a usage change for [ARC] LIST

The [cover] [feature class] and {NONE} parameters are no longer used by the [ARC] LIST command. Instead, specify the full name of the file you want to have listed, including its suffix.

LIST [info_file] {range} {item...item}

Continued support for concatenated items and item ranges

In situations where a Character type item is accepted as an argument on a command line, a concatenated item can be used. Item concatenation is achieved by using the following syntax:

Begin_Item+End_Item

This syntax convention indicates that Begin_Item and End_Item, and all items between them should be used as one single Character item. Concatenation is temporary and is only in effect during command processing. Note that blank characters are valid characters and must be accounted for during item comparison and queries.

Item ranges can be processed in reverse order

Some commands provide an option to specify multiple items on a command line. In these situations, PC ARC/INFO provides a shorthand convention for listing them:

Begin_Item: End_Item

This indicates that Begin_Item and End_Item and all items between them will be used as if they were individually listed on the command line. With PC ARC/INFO 4.0, item ranges can be submitted in reverse order. This means that if End_Item comes before Begin_Item in the database file, the items will be processed in reverse order.

New L command options

The L command lists information about files and directories. Several new options have been added and a new command-line parameter allows the output to be directed to a file rather than to the screen. (Depending on your computer's command processor, DOS output redirection (>) may not be available.) Additionally, the -LP option has been modified to list polygon coverages (PACKed coverages no longer exist) and the -D option now outputs with a four digit date. The version 4.0 L command lists double precision coverages unless the new -L1 option is used.

L {path...path} -{options} { [-WA / -WI] [output_file] }

New and modified options:

? Print usage and options only
D Print date modified. Output is in the format MM/DD/YYYY
X Print extended long file name
OB Sort entries by backwards date and time
LC List double precision coverages
L1 List single precision coverages
LP List polygon coverages only
LA List arc coverages only
LX List point coverages only
LS List address coverages only
LT List annotation coverages only

-WA [output] Write Append - append the information to an existing [output] file.

-WI [output] Write Initialize - write the list to a new [output] file.

(Note the change to the LP option. In previous versions of PC ARC/INFO, LP listed PACKed coverages. Since the DOS PACK command no longer exists, LP now lists POLYGON coverages.)

LISTCOVS supports both single and double precision coverages

As with the L command, LISTCOVS has a new parameter that allows you to search directories for single precision coverages from earlier versions of PC ARC/INFO. The default listing of LISTCOVS is double precision coverages.

LISTCOVS {workspace} {DOUBLE / SINGLE} {output_file}C

Note that you can use the number 2 to denote DOUBLE precision or the number 1 for SINGLE precision wherever these keywords are used.

For example:

LISTCOVS	Lists all double precision coverages on the current directory
LISTCOVS \mydir SINGLE	Lists all single precision coverages on \mydir.

Selection Sets can be created at ARC as well as in ARCEDIT and ARCPLOT

Sets of selected records can be created at the ARC processor level as well as in ARCEDIT and ARCPLOT.

Usage for ARC processor selection commands:

SELECT [info_file] {logical_expression}

Creates a new set of selected features of a database file. Specify the name of the file [info_file] and the expression {logical_expression} that represents the record(s) you want selected. Entering only the database file name causes all records to be selected.

ASELECT [info_file] {logical_expression}

Adds features to the selected set of a database file. Specify the name of the file [info_file] and the expression {logical_expression} that represents the record(s) you want selected. Entering only the database file name causes all records to be selected.

RESELECT [info_file] {logical_expression}

This command selects a set of features from the selected set of a database file. If {logical_expression} is not entered, the selection set is cleared leaving no selected records.

NSELECT [info_file]

Switches the selected and unselected sets of the specified database file. Those records of [info_file] that were unselected now become the selected ones.

UNSELECT [info_file] {logical_expression}

UNSELECT causes the record(s) specified by {logical_expression} to be removed from the selected set of [info_file]. If only the database file name is entered, the selected set is cleared leaving no selected records.

CLEARSELECT {info_file}

Clears all selection sets from the current workspace, or the selection set of the specified {info_file}.

LISTSELECT

Lists all selection sets for all data files in the current workspace.

READSELECT [info_file] [selection_file]

Selection sets saved with the WRITESELECT command can be restored in ARC, ARCEDIT and ARCPLOT with the READSELECT command.

WRITESELECT [info_file] [selection_file]

A selection file is created with WRITESELECT. It is a description of a previously selected set of records for a data file. READSELECT restores the data file's selected set.

In addition, these commands now use the current set of selected records rather than using sub-commands to establish the set of records to be processed:

ELIMINATE EXTRACT

Selection Sets can be used between PC ARC/INFO modules and sessions

The commands WRITESELECT and READSELECT are new for all PC ARC/INFO modules. WRITESELECT saves a selected set of records to a selection file. READSELECT reads a selection file and restores the selected set.

They allow selected records from a coverage attribute table or other data file to be saved and used at a later time. Selection sets can be written and read in every module. For example, a set of features can be selected using ARCPLOT's selection tools and written to a selection file using WRITESELECT. This set of records can then be restored in a subsequent ARCEDIT session with the READSELECT command. The selection set can also be used in an entirely different ARC session. Multiple selection set files can be written for one or more data files and can be kept indefinitely.

The RELATE command replaces the TABLES JOIN command

```
RELATE ON      [from_info_file] [to_info_file] [from_item] {# / to_item}
               {LINEAR / ORDERED / TABLE / LINK}
RELATE OFF     [from_info_file] {to_info_file}
RELATE LIST    [from_info_file] {out_file}
RELATE HIDE    [from_info_file]
RELATE SHOW    [from_info_file]
RELATE TEST    [from_info_file]
```

In addition to the many RELATE options, there is another significant difference between the previous TABLES JOIN command and the new RELATE command. Note the {# / item} parameter on the RELATE ON command line. The item in [to_file] that is used to form the RELATE does not have to be identical to the item on the [from_file]. It only needs to be the same type (Numeric or Character).

RELATE ON - sets up a relate between two data files.

RELATE OFF - removes relates between data files.

RELATE LIST - lists relates on the screen or to a file.

RELATE HIDE - temporarily suspends (hides) an established relate.

RELATE SHOW - re-establishes a previous relate that was hidden.

RELATE TEST - tests the relations established for a data file. Any errors will be displayed on the screen.

The following commands will automatically establish, and use, existing RELATE(s) when processing coverage data files:

ARC ARCATLAS ARCDIME ARCDLG ARCDLGN

ARCDXF	ARCEDIT	ARCIGES	ARCMOSS	ARCPLOT
BUFFER	CALCULATE	CLEAN	COPYINFO	COPYCOV
DISSOLVE	FREQUENCY	GENERALIZE	INDEXITEM	INFODIRECTORY
ITEMS	KILL	KILLINFO	LINEGRID	LIST
MOVEITEM	NEAR	POINTDIS	POINTGRID	POLYGRID
PROJECT	PULLITEM	RELATE	RENAMCOV	SORTREC
SPLIT	STATISTIC	TIGERTOOL		

...and all SELECTION commands in all modules (SELECT, RESELECT etc.)

For example:

Note that PC ARC/INFO refers to related items by preceding their name with a '#' sign.

ARCDXF - looks for "#DXF_" items in any related file in addition to "DXF_" items in the PAT and AAT.

ARCDLG / ARCDLGN - look for #MAJOR / #MINOR attribute pairs on any related file in addition to MAJOR / MINOR attribute pairs in the PAT and AAT.

ARCIGES - looks for the related item "#IGES_" in any related file in addition to the "IGES_" items in the PAT and AAT.

ARCDIME - items that are searched for in the AAT are also searched for in all related files.

Since RELATE is an [ARC] level command, the ACODE, XCODE or PCODE files created by the following commands can be RELATED to the coverage attribute tables rather than having to be permanently written to them using JOINITEM:

DIMEARC	DLGOARC	DLGSARC
DXFARC	ETAKARC	IGESARC

POINTDISTANCE automatically sets up RELATEs between [info_file] and the two point coverages, if two coverages are specified. If there is only one coverage, the RELATE is not established.

FREQUENCY automatically sets up a relate from the [in_file] to the [out_file] on the case item, if a case item is specified.

NEAR automatically sets up a relate from the PAT of [point_cover] to the PAT of [near_cover] (if the POINT option was specified) or to the AAT of the [near_cover] (for the LINE option). The relate item is [cover_item] and LINK is the method.

TIGERTOOL, a new command, automatically sets up numerous relates. It will join the line cover ACODE file to the AAT file and then remove the ACODE file. It will join the point cover XCODE and TYPEA files to the line cover PAT file and then remove the XCODE and TYPEA files. If there is a landmark cover it will join the landmark XCODE file to the landmark PAT file and remove the XCODE file. The following relations are also set up:

- AAT to TYPE3 using TLID
- AAT to TYPE5 using TLID
- AAT to TYPE6 using TLID
- TYPE6 to TYPEZ using TLID+RTSQ
- AAT to TYPEH using TLID
- PAT to TYPE9 using CENID+POLYID
- PAT to TYPES using CENID+POLYID

Permanent multilevel RELATES

When you enter RELATE ON, the relate information ([to_info_file], [from_item], {to_item} and {method}) is written to a file in the coverage directory named for the [from_info_file] with a .REL suffix. For example, a relate established between a coverage PAT file (*cover.PAT*) and a lookup table (*cover.LUT*) would create the file *cover\PAT.REL*. You can relate the [from_info_file] file to more than one [to_info_file] file by issuing multiple RELATE ON commands. After the first RELATE is established and a .REL is created, the information about each subsequent relationship established for [from_info_file] is appended to the .REL file. This allows you to have several relates operating at once. Additionally, the relate information stays permanently with the [from] file until you remove it using the RELATE OFF command.

When a data file is opened, a search is performed for a matching relate file (.REL). If one is found, it is also opened and all [to] files are joined to the [from] file. If the [to] file also has a .REL file, those relations will also be recreated. In this way multi-layer relates can be established. All items and related items of the data file are available to PC ARC/INFO commands. As in all previous version of PC ARC/INFO, related items are referenced by preceding their name with the '#' sign.

The following command lines will establish two RELATES. One to the PAT and another to the AAT of the coverage MYCOVER:

```
[ARC] RELATE ON mycover.pat parcels item (creates mycover\pat.rel)
[ARC] RELATE ON mycover.aat symbols item (creates mycover\aat.rel)
```

These RELATES will be automatically established any time the PAT or the AAT of the coverage is opened. In ARCPLOT then, you can draw coverage features based on an item of the related file. (Note the use of the # sign.)

```
POLYGONSHADES mycover #class (uses the values of the item 'CLASS' from
                              PARCELS.DBF)
ARCLINES mycover #surface (uses the values of the item SURFACE from
                           SYMBOLS.DBF)
```

This RELATE information is preserved between sessions, so once you set a relate (RELATE ON), it stays that way until you turn it off (RELATE OFF), temporarily hide it (RELATE HIDE) or establish a new RELATE ON. You can re-establish a previous relate that was hidden (RELATE HIDE) by using the RELATE SHOW command.

A file can also have more than one file RELATED to it. Each RELATE is appended to the .REL file:

```
RELATE ON mycover.pat parcels item1
RELATE ON mycover.pat names item2
```

You can also establish more than one level of RELATE. For example:

```
RELATE ON mycover.pat parcels class
RELATE ON parcels landuse shade
```

Now, whenever MY.COVER is used, the RELATE to PARCELS will be established and since PARCELS has been opened, the RELATE between PARCELS and LANDUSE will also become active. This establishes a connection between MY.COVER and the file LANDUSE.

RELEDIT - A Windows interface for RELATE

Relates can be viewed, and managed interactively with the RELEDIT command.

INFODIRECTORY will list the RELATE status of each of the database files

INFODIRECTORY {path} {BRIEF / LIST / LONG} {ALL / ON / OFF}

{path} - the pathname of the files to be listed. Wildcard names are permitted.

BRIEF - file names are listed four per line

LIST - file names are listed one per line

LONG - the files are listed one per line with four columns of information: the file name, number of records, record length and relate status. There is one header line at the beginning of the list with the titles of the columns.

ALL List all files whether their RELATE status is "ON" or "OFF".

ON List only those files whose RELATE status is "ON".

OFF List only those files whose RELATE status is "OFF".

Example:

INFODIRECTORY # LONG ON

DATA FILE NAME	NO. RECS	LENGTH	RELATE
STATES.DBLTIC	4	52	OFF
STATES.DBLBND	1	81	OFF
STATES.PAT	8	132	ON
STATES.AAT	87	101	OFF

To direct this listing to an output file use the &OPENW directive. For example:

&OPENW list.fil

INFODIR # LIST

&CLOSEW

INDEXITEM creates a new index file for indexed items

INDEXITEM creates an attribute index file on an item in a database data file to increase access speed to the specified item during query operations. The new usage is:

INDEXITEM [info_file] [item]

[info_file] - the name of the data file containing the item to be indexed.

[item] - the name of the item to be indexed. It may not be a related or combined item.

INDEXITEM now creates one index file for each info file. Any number of items can be indexed from [info_file], but only one DBX file is generated per [info_file]. A file named *info_file*.DBX is placed in the same directory as [info_file]. So, if [info_file] is PAT.DBF, an index file named PAT.DBX would be created. If [info_file] is modified, the index file will be deleted. If an operation outside of PC ARC/INFO .modifies the info data file, the next time an attempt is made to use the index file, it will automatically be deleted because it is out of sync.

Indexed items are now automatically used with certain simple logical expressions

Logical expressions of the following form will use an indexed item, if it is entered. Note that the IN operator, although still accepted, is no longer required when forming the expression and the '\$' sign is no longer required as part of the item name:

[item] [EQ | NE | GT | GE | LT | LE] [value]

or
[item] IN {...}

An Indexed item can be used as either operand1 or operand2 of a logical expression but not both in the same expression. (A logical expression is expressed as [operand1] [operator] [operand2].) The other operand must be a constant. Additionally, only numeric operators may be applied to an indexed item. You cannot use an indexed item with CN or NC

Note that for most logical expressions, if an item has been INDEXed, the index is automatically used. Previous versions of PC ARC/INFO required that indexed items be prefixed with a dollar sign (\$), when using the IN operator. Version 4.0 does not use the dollar-sign notation.

INDEXEDIT is a new item index file management tool

INDEXEDIT is an interactive editor that displays the items of a data file and lets you add indexed items to its index file or delete the index file. KILLINDEX can also be used to remove index files.

KILLINDEX can be used to remove an item index file or the spatial index file

```
KILLINDEX [info_file] {INFO}  
KILLINDEX [cover] [COVER]
```

If you specify INFO on the command line, KILLINDEX will assume that the 1st parameter is an info file and it will try to remove the index (.dbx) file for it.

If you specify COVER, it will delete the spatial indices for the specified coverage. These includes the XBN, XBX (point), ABN, ABX (arc), PBN, PBX (poly), TBN, TBX (default annotation) and *.TBN, *.TBX (annotation subclass) files.

New file format for the output from UNLOADANNO

Improvements to the way PC ARC/INFO 4.0 displays and manages annotation features require changes to the UNLOADANNO command and to the files it creates. First, the command now supports annotation subclasses. (Creating and using subclasses is discussed in the ARCEDIT section of this file.)

```
UNLOADANNO [cover] {subclass} {#_points}
```

If {subclass} is not entered, the coverage 'default' annotation is copied to a database file. It is named [cover].def_anno. Subclass annotation is written to a database file named [cover].{subclass}.

{#_points} is also a new option and represents the number of XPOINT and YPOINT items to reserve in the database file for shape points. The maximum number that can be stored is 85. Since annotation can be defined by up to 500 points, annotation that exceed {#_points} will have their shape points weeded down.

The structure of the output file from UNLOADANNO also differs from 3.5.2. This means that output created by UNLOADANNO in previous versions of PC ARC/INFO cannot be used with the 4.0 version of LOADANNO and vice versa. A description of the new file structure is included in the online command reference for UNLOADANNO.

LOADANNO supports annotation subclasses

LOADANNO copies records from the database files created by UNLOADANNO as annotation into a coverage. Default subclass annotation is copied from [cover].def_anno and individual

subclass annotation from their [cover].{subclass} files. It is used to load annotation edits made using TABLES or your database manager back into coverage annotation. Annotation subclasses are discussed later in this guide.

LOADANNO [cover] {subclass}

If {subclass} is not entered, or is entered as #, the coverage 'default' annotation is copied from the database file [cover].def_anno to the default annotation subclass. {subclass} annotation is copied from the [cover].{subclass} files to their appropriate annotation subclasses.

LOADANNO will replace the existing annotation for each specified subclass of the coverage. It will not save any annotation to the subclass except those that are represented in the database file. Remember that the files created by UNLOADANNO at version 4.0 are structurally different from previous versions of PC ARC/INFO (reflecting subclass support). This also means that LOADANNO can only read files created by UNLOADANNO 4.0.

You can now specify a background color for all text windows using the COLOR and DIALOGCOLOR commands

COLOR [foreground_color] {background_color}

DIALOGCOLOR [foreground_color] {background_color}

Note: COLOR is only available at the [ARC] prompt. DIALOGCOLOR is available in all modules.

DIALOGCLEAR has been added to the ARC module

To clear the text window at the [ARC] prompt, enter the command name DIALOGCLEAR. DIALOGCLEAR continues to be supported in ARCEDIT and ARCPLOT.

PC ARC/INFO command lines can include up to 1024 characters

Command lines can include up to 1024 characters. &ASK and &RESPONSE also display 1024 characters.

WinTab and DIGFORM digitizer support

WinTab drivers can be specified by using the CONDIG command:

CONDIG [wintab_digitizer]

There are two DIGFORM files for WINTAB digitizers: WINTAB4 and WINTAB16.

Note that the DIGFORM subdirectory is still part of PC ARC/INFO 4.0, so all previous Format Files remain available for use. There are also new format files in DIGFORM to support the SUMMAGRID V digitizer (SUMGRID5) and the SUMMAGRID IV digitizer (SUMGRID4).

The following commands now display Date items using 4 digits instead of 2

ARC: LIST and L
ARCPLOT: LIST and IDENTIFY
ARCEDIT: LIST

STATISTICS command at [ARC]

In previous version of PC ARC/INFO, statistics could be generated for an item in a database file in ARCPLOT. This utility has been added to the ARC processor, as well.

STATISTICS [info_file] {item} {case_item}

The STATISTICS command will compute the COUNT, MIN, MAX, SUM and MEAN for all selected records of [info_file]. The screen output can be saved to a file using the SML directive &OPEN before issuing the STATISTICS command. The summary total can also be stored to SML variables using the SHOW command.

CLASSIFY is a new interactive version of the CLASS command in ARCPLOT

CLASSIFY [cover] [ARC / POINT / POLY]
CLASSIFY [info_file] {INFO}

CLASSIFY displays a window in which possible classifications can be modeled, viewed and saved. Saved classifications can be used as code values to link items together or written to a Legend file for use in ARCPLOT.

New TIGERTOOL command

TIGERTOOL is a command that converts TIGER line, point and landmark coverages created by TIGERARC into a polygon coverage containing all the associated line and geographic area attributes and a point coverage for file sets containing landmark point features.

TIGERTOOL [line_cover] [point_cover] {landmark_cover}

[line_cover] - The line coverage created by TIGERARC. All intersecting arcs must already be split.

[point_cover] - The point_coverage created by TIGERARC. Note: This coverage will be deleted once the TIGERTOOL operation is successful.

{landmark_cover} The landmark coverage created by TIGERARC.

TIGERTOOL should only be run on coverages created with TIGERARC. You risk losing data if it is used with other coverages.

Before using TIGERTOOL, first run TIGERARC. Then use CLEAN with the LINE option on the resulting line coverage to identify, and split, all intersecting arcs in the coverage. Once the line coverage is clean, run TIGERTOOL on either the line and point coverages or the line, point and landmark coverages.

TIGERTOOL performs many operations. It should be noted, that with large coverages, it may take a while to complete. TIGERTOOL adds polygon location points from the point cover to the line cover. It builds both line and polygon topology for the line cover and copies all attribute tables associated with area features to the line cover. After the point cover information is written to the line cover the point cover is deleted.

It will join the line cover ACODE file to the AAT file and then remove the ACODE file. It will join the point cover XCODE and TYPEA files to the line cover PAT file and then remove the XCODE and TYPEA files. If there is a landmark cover it will join the landmark XCODE file to the landmark PAT file and remove the XCODE file.

The following RELATEs are also established with [line_cover]:

AAT	to TYPE3 using TLID
AAT	to TYPE5 using TLID
AAT	to TYPE6 using TLID
TYPE6	to TYPEZ using TLID+RTSQ
AAT	to TYPEH using TLID
PAT	to TYPE9 using CENID+POLYID
PAT	to TYPES using CENID+POLYID

New SHAPEDXF command

SHAPEDXF converts an ArcView shape file into a DXF file or displays information about a shape file.

```
SHAPEDXF [shape_file] [dxf_file] [decimals]
SHAPEDXF [shape_file] {output}
```

[shape_file] - The name of the shape file to convert. Alternatively, you can obtain information about [shape_file] by entering it alone on the command line or with the {output} option.

[dxf_file] - The name of the DXF file to create.

[decimals] - The number of decimal digits to use when writing coordinates and other non-integer numbers in the DXF file. It must be a number from 0-9.

{output} - The name of the text file to write shape file information to if [dxf_file] and [decimals] are not entered on the command line. If no output file is specified, the output goes to the screen.

Notes:

Point shape files are written out as point entities.

Line shape files are written out as unclosed polyline entities.

Polygon shape files are written out as closed polyline entities.

During the conversion, additional attribute information about lines and points is extracted from the shape file if the following items are present:

LAYER - a character item which specifies the layer name. The default layer will be SHP_POINT, SHP_LINE or SHP_POLY depending on the type of the shape file.

COLOR - an item containing the color of each DXF entity. The default color is white (7).

ELEVATION - the Z value of the entity. The default is 0.0.

THICKNESS - a numeric item whose value is the thickness of the entity. The default is 0.0.

Quoted strings on command lines can be enclosed with either single or double quotation marks

All command lines that used single or double quotes to specify the beginning and ending of text strings can now use either single quotes or double quotes. However, you must start and end the string with the same type of quote (either single or double). This makes it much easier to specify a single quote or double quote in a string. A single quote can appear within a double quoted string and a double quote can appear within a single quoted string without having to duplicate it.

All 'External' commands now return an error code to indicate successful or unsuccessful command processing

All PC ARC/INFO External commands return an error code upon completion. (External commands are discussed earlier in this guide.) Your SML applications can now include a check for successful command processing by capturing the command's return code. In this way, if a coverage is going to be processed by several commands, you can trap for unsuccessful processing by one command before passing it on to another. A returned value of '0' indicates successful command processing. If unsuccessful, the returned value is non-zero. Use the &RV directive to retrieve the error code. For example:

```
CLEAN cov POLY
&RV -1
&GOTO ERROR &GT %-1 0
```

The results of an &SYSTEM program can also be retrieved with the &RV statement.

A returned value of '0' indicates successful processing. If unsuccessful, the returned value is non-zero.

```
&SYS program_name
&RV -1
&GOTO ERROR &GT %-1 0
```

SML Enhancements

Complex logical expressions are now supported in SML

&AND and &OR can be used to combine logical statements on one line.

The block-structured &IF statement can now be used in SML and CML macros

Block IF conditional statements are now supported by SML macros and not only in compiled CML macros.

```
&IF [logical_expression] &DO
    {true_code}
{&ELSEIF [logical_expression] &DO
    {true_code}
{&ELSE
    {false_code}
&END
```

A new '&IFNOT' conditional statement has been added to SML and CML logical processing

&GOBACK, &GOTO and &JUMP directives include a conditional statement that determines whether to branch to another location in the macro. This statement can now include an &IFNOT logical expression in addition to &IF.

i.e., &GOTO [label] {&IF/&IFNOT [logical_expression]}

When &IFNOT is encountered, processing will jump to [label] if the statement is FALSE. &IF statements will continue to jump to [label] if the statement is TRUE.

Many new logical operators have been added to SML and CML logical processing

The following logical operators are now available for use within a [logical_expression]:

&COV and &NCOV

&IN and &NIN - determine whether a value is in (&IN) or not in (&NIN) a list of values. Note that the comparison is case insensitive. Usage: &IN value "list"

&ITM and &NITM - determine whether an item exists (&ITM) or doesn't exist (&NITM)

&GE and > - determine whether a value is greater-than-or-equal to a value (&GE) or greater-than a value (>)

&INFO and &NINFO

&LE and < - determine whether a value is less-than-or-equal to a value (&LE) or less-than a value (<)

&OPH and &NOPH - check to see if a file handle is open (or in use) by &OPEN or &OPENW. Usage is: &OPH {file_handle}. Enter the value 1,2 or 3 for {file_handle} to check on an &OPEN statement. Values -1, -2 and -3 will check for an &OPENW file handle.

&OPI and &NOPI - check to see if a database file is open, is in use or if some or all of it is read only.

&OPN and &NOPN - check to see if a file, directory or coverage is open, is in use or if some or all of it is read only. For example: &if &opn %cover% &do

New Arithmetic Operators

There are five new arithmetic equation functions that can be used with &CALCVAR or with embedded expressions.

Four of them are used to get information about an item or related item. The function name appears first, followed by the data file name, and then the item name. If the item is not found, the functions return a value of -1.

ITMD - returns the number of decimal places of a numeric item on a data file. If the item is memo, date, character or integer, the returned value is zero.

ITMP - Returns the byte position of an item on a data file. If it is a related item, the value will be less than -10000. The leftmost digit indicates what related file its on (first, second third etc). The right four digits indicate its position. The position is calculated by taking the sum of widths of the main data file and any lower related files plus the byte position of the item in its data file.

ITMT - returns the type of an item on a data file. The value returned is a number from 0-4.

0 = Memo item.

1 = Date item.

2 = Character item.

3 = Integer item (no decimals)

4 = Numeric item (decimals)

ITMW - returns the width of an item on a data file.

A fifth arithmetic function is used to get information about a file or a directory.

FTYPE [name] - returns the following values:

0 = [name] is a file
1 = [name] is a directory
-1 = [name] is not found

New SML commands

&ARCPROG - runs a PC ARC/INFO executable program located in the ARCEXE\PROGRAMS directory. There are a number of "DOS" functions that can be accessed via the &ARCPROG function (ren, mkdir, del, deltree, rmdir, etc). These run with no error messages.

&WIN - is an SML version of the WIN command with two differences. As an SML command, it is not recorded when using "&ECHO &COM" but it is when using "&ECHO &ON". The WIN command is recorded with both "&ECHO &COM " and "&ECHO &ON ". Secondly, the result of the &WIN command is not returned in SML variable 1. Instead, you assign the result to a variable of your choice by using the &RV command.

```
&WIN FILE C * "Pick a coverage."  
&RV -10  
&TYPE "The coverage is %-10"
```

&WS - changes workspaces or saves a workspace location to an SML variable.

&IF - support for block IF statements (described above).

&VAR - support for named variables (described below).

Named variables are now supported in SML by specifying them with the new &VAR statement

```
&VAR [var_name] [var_number] {[var_name] [var_number] ...}  
&VAR &DEL [var_name / var_number / &ALL]
```

The value of [var_number] will be assigned to [var_name] for that routine. Up to 50 variables per routine can be named.

The Arc WIN command has a new SML companion command named &WIN

The multifunction &WIN and WIN commands have many KEYWORD options that provide a suite of functions for interfacing with Microsoft Windows. &WIN is new for version 4.0.

These two commands are nearly identical. The main difference is that the WIN command returns its values to SML variable 1, while the &WIN command returns its values via the &RV statement. For this reason, using WIN may be more convenient when working interactively from the keyboard. However, &WIN provides more control in an SML application because you can direct its output to a particular variable number. For example:

WIN Command	&WIN Command
WIN PATH .	&WIN PATH .
&rem Result in SML variable 1	&rv -11
&type "%1"	&type "%-11"

Another difference between WIN and &WIN is how &ECHO &WATCH behaves. Since &WIN is an SML command, its output is not recorded to the output file when you are using &ECHO &WATCH to record a session.

There have been many enhancements made to the WIN command, and to its counterpart &WIN. Some of the significant changes are noted in this guide, but you should refer to the on-line command references for details on using these tools.

Bitmap images can be displayed on widgets now as well as DIS files (BRECT, WRECT, GFRM etc.). In addition, the BRECT, WRECT widgets have new options that will fix the color of the rectangle or allow the color of the rectangle to match the background color of the graphics screen. UBUT will vary whether black or white is the foreground color based on the current graphics screen setting.

Some enhancements:

WIN COLOR [* / [R / N / W] [file] / V [color]]

WIN PATTERN [[R / N / W] [file] / V [pat]]

WIN DB [C / R / D / P / U / BH / BI / BD / E] {file} {dialog#}

WIN FILE [1-3/ D / N / G / I / R / C / S / T / A / P / X / M / O] [suffix(es)] [title] {P / F}

And new WIN EXEC parameters. For example:

WIN EXEC 9 {P} - returns a 5 digit font number (fffff). If {P} is used, an 8 digit number is returned (pppfffff). The first 3 digits represent the pattern number (32 - 255) of the selected symbol from that font.

The command &WIN FILE has a new parameter controlling the length of the returned pathname as well as revised options for listing files and coverages

&WIN FILE displays the standard Windows file picker and returns the selection to an SML variable.

&WIN FILE [1 / 2 / 3 / I / R / G] [suffix(es)] [title] {P / F}

&WIN FILE [D / C / S / T / A / P / X / N / M / O] [*] [title] {P / F}

[1 / 2 / 3 / I / D / A / M] - unchanged options

[C / A / P / X / S] - these options now list version 4.0 (double precision) coverages

[G / N] - these options now ask if it is okay to overwrite an existing data file (G option) or an existing coverage (N option). Previously, they wouldn't let you enter the name of an existing file or coverage. The N option also displays both single and double precision coverages.

[R / T / O] - new options for version 4.0

R - allows the user to pick data files that have an active RELATE

T - allows the user to pick a double precision annotation coverage

O - allows the user to pick an "old" or single precision coverage

{P / F} - controls whether a partial or full pathname of the selected file/coverage will be returned.

P - if the file, coverage, or info file is picked from the current directory or any subdirectory of the current directory only the relative path is passed back and not the full pathname. If it is from a different directory path, then the full pathname is passed back.

F - a full pathname is always returned

&WIN IDLE now has a {seconds} option

&WIN IDLE {seconds}

Specifying a value for {seconds} yields that period of the Windows time slice.

&WIN RUN / RUNW return new values upon command completion

&WIN RUN launches an application and immediately returns control to the launching text window. The SML variable will have one of the following values upon completion:

- 1 - the application did not launch
- 0 - the application launched okay

WIN RUNW launches an application and returns control to the launching text window after the application has completed. The SML variable will have one of the following values upon completion:

- 1 - the application did not launch
- 0 - the application launched and ran okay
- >0 - the application launched, but the program returned an error (DOS errorlevel)

There are now more WIN DB Dialog Definition File Statements and Dialog File Commands. These add more user control to the look of on-screen dialog boxes.

Dialog File Commands

Add "H" and V dialog commands. These widget commands make widget(s) visible (V) or hidden (H). Either one or a range of widgets may be specified. A hidden widget can not be seen and it can't be chosen by the user.

V [begin_widget_id] {end_widget_id}
H [begin_widget_id] {end_widget_id}

Note that you are now able to specify a range of widget-ids. This is true for the C, U, G, and E commands as well. This simplifies and speeds the processing of a large range of widgets.

Dialog File Statements

A) Four new Frame widgets have been added:

Frame widgets draw 3D frames on a dialog box. These new frame widgets include options that allow you to change the appearance of the frame as well as choose which sides of the frame to draw:

RFRM (raised frame) - looks like a button
SFRM (sunken frame) - looks like an inverted (pushed in) button
EFRM (etched frame) - looks like a groove
LFRM (lipped frame) - looks like a small ridge

Each of these has the same syntax. The following example uses the RFRM widget:

RFRM [widget_id] [row] [col] [mrow] [mcol] v {options}

These widgets must have a nonzero [widget_id] to produce the best drawing effect. The widgets are always non-pickable. If the [widget_id] is zero, the options will be ignored and frame will be a Sunken or Etched frame. A picture can not be placed on them like the other frames and rectangles. The options can be changed with the "S" dialog file command.

Any combination of the following options is allows you to control which sides of the frame will be drawn. If none are specified, then all edges of the frame will be drawn:

- T - Draw the top of the frame.
- B - Draw the bottom of the frame.
- L - Draw the left side of the frame.
- R - Draw the right side of the frame.

Any combination of the following additional options allows you to further control the appearance of the frame:

- S - Soft edges instead of harder "tiled" edges.
- M - Mono or one dimensional border.
- F - Flat border.

B) A Tabbed Button widget has been added:

TBUT [widget_id] [row] [col] [mrow] [mcol] [V / F / VO / FO] {text, begin_id end_id}

This is a tab button for creating a tabbed dialog box. Several tab buttons are placed on the dialog box with the same row position and with no space between each button. Each tab button has a set of widgets associated with it. Pushing a tab button will cause it to be drawn as a foreground tab button and its widgets will be made visible. The other tab buttons on the dialog box will be drawn as background tab buttons and their widgets will be hidden.

Number of local variables increased to 50

There are now 50 local variables available for use with SML. This is an increase from 20. They are numbered -1 through -50.

SML variables can include strings up to 256 characters

SML variables can be up to 256 characters long. This is an increase from 80.

Edit boxes can have input strings up to 256 characters

SML variables -50 through 9999 can now contain strings with up to 256 characters. An edit box created with the WIN DB command can also handle input of up to 256 characters.

&LISTVAR will now list variable values in either ascending or descending numeric order (including the new SML named variables). It can also provide information about named variables

&LISTVAR {begin_var} {end_var}
&LISTVAR &VAR

Variables can now be listed in ascending or descending numeric order depending on the order in which the variable numbers are entered on the &LISTVAR command line. If {begin_var} is less than {end_var}, the listing is in ascending order. If {begin_var} is greater than {end_var}, the listing is in descending order. If {end_var} is not specified on the command line, &LISTVAR

displays the variables in ascending order. This is particularly useful when you want to display the local variables in the order -1 through -50. Named variables can also be used.

&LISTVAR &VAR - Lists all named variables, their numbers and values.

&R is a new version of &RUN

Users are advised to use &R rather than &RUN or @. &R writes the variables that are being passed into the program into local variables rather than global ones. This ensures the integrity of global variables throughout an application. Note that you should also use &RETURN and &RV to return values to the calling routine.

&RUN remains an SML directive because many User applications still exist that use only global variables. In practice, new applications should pass values between routines through local variables with the &R directive.

Developer's macros now use the new &R directive to execute. &RUN and @ will no longer work with them

The SML Developer's Toolkit includes many useful macros that you can use in your applications. They are located in SCR\TOOLS on the ARCEXE directory and are described in the 'SML Developers Toolkit' section of the on-line Help. If you have used them previously in your applications, you should note the following:

These macros now use the new &R directive to run. Previously, they were run with &RUN or @.

The toolkit macros that used outdated utilities have been removed. Some have been added and all of them have been updated to use Windows tools. These macros may work differently than they did in previous versions of PC ARC/INFO. Refer to the beginning of this guide for a list of new and removed commands.

&KEY has a new usage

The &KEY directive (which assigns the first keystroke or function key number to a variable) no longer supports the &NW option. &NW was a DOS feature that could be used to check for user input without displaying a message. The new format of &KEY is as follows:

&KEY [var] [string]

&ASK and &RESPONSE can display up to 1024 characters

&ASK and &RESPONSE can include up to 1024 characters. PC ARC/INFO command lines can also display 1024 characters.

In order to maintain numeric precision, the results of some calculations performed by &CV will be stored in scientific notation

Fifteen (15) digits of accuracy can now be stored in SML variables. If the absolute value of a variable exceeds that degree of accuracy (i.e., <1E-15 or >= 1E15), it is stored in scientific notation:

```
&CV 1 1234567890123450
&CV 2 12345.6789012345
&CV 3 0.0000123456789012345
```

```

&CV 4 0.00000123456789012345
&CV 5 123.45

&LV 1 5
VAR. VALUE
%0001 0.123456789012345E+16
%0002 12345.6789012345
%0003 0.0000123456789012345
%0004 0.123456789012345E-05
%0005 123.45

```

Changes affecting both ARCPLOT and ARCEDIT

It is now possible to pass command parameters into ARCPLOT and ARCEDIT on start-up

```

[ARC] ARCPLOT { sml_file {command_arguments}}
[ARC] ARCEDIT { sml_file {command_arguments}}

```

An optional SML or CML file containing ARCEDIT or ARCPLOT commands and SML directives can be specified on the ARCPLOT and ARCEDIT command lines. It will be executed after the program is started. If a *module.INI* file exists, it is executed as part of the startup and is processed before {sml_file}. {sml_file} may also be any External ARC command. Initialization files and External commands are discussed earlier in this guide.

DISPLAY no longer clears the text screen

Entering the DISPLAY command will clear the graphic screen, but not the text screen. Use DIALOGCLEAR to clear the text screen in ARCPLOT and ARCEDIT.

SCREENSAVE creates a new display list file format

The display list files between 3.x and 4.0 are not interchangeable. SCREENRESTORE will not be able to restore a display list file created in earlier versions of PC ARC/INFO.

MEMO items no longer supported

[ARC] LIST and other coverage management commands that copy data files will retain existing MEMO items, but they are not used by PC ARC/INFO 4.0.

SYMEDIT - a new interactive Symbol Editor

SYMEDIT is a new graphic command that allows interactive symbol editing in both ARCPLOT and ARCEDIT. You can now edit any or all attributes of the symbols in the current line, marker and shade sets. Additionally, you can edit colors and hardware shade patterns. This new editor replaces, and expands, the older version line symbol editor LINEEDIT and the SML tools named COLORS and PAT.

Note that once the command has been entered, a dialog box will appear in which you can specify individual values for each attribute of that symbol. Up to three drawing layers can be defined for shade and line symbols. You may choose any number of symbols to edit without leaving the dialog box. Changes you make can be saved to a new symbolset for use in later sessions or can be applied to the current session only.

SYMPICK - a new interactive symbol picker

Symbols can be selected from a screen display. You can choose to:

1. Pick a new current symbol set.
2. Pick a new current symbol from the current symbol set.
3. Pick a symbol from any of the available symbol sets and copy it to the current symbol set.

Menu options available when using the mouse

There are many commands in ARCEDIT and ARCPLOT that require a number to be entered to signal a specific graphic action. For example, when digitizing arcs in ARCEDIT, pressing a 2 button signifies a node to be entered, pressing a 1 button adds a vertex and pressing a 9 button signifies the end of input. When using a digitizer to identify locations, these buttons are found on the digitizer's cursor. When using the screen cursor to point to locations you can use either the keyboard number keys or the buttons on your mouse. The following list identifies which mouse buttons send which value:

Mouse button	Value returned
Left	1
Right	2
Left and Right	9

Right - Held down for ¼ second:

The right mouse button will also display a full menu of all available choices for the operation you are performing if you hold it down for more than ¼ second. In this way you can select an option for your drawing command from the mouse submenu rather than from the keyboard. When a drawing command is not active, holding the right mouse button down in the graphic window will display the 'Windows' Pan and Zoom options.

And if you have a three button mouse:

Middle	4
Middle and Left	3
Middle and Right	6
All three	8

New symbol file options for LINESET / MARKERSET / TEXTSET / SHADESET allow symbol files to be altered and created

New usages:

```
LINESET [* / [lineset_file] {start_sym} {end_sym} {new_start#} {CLEAR}]
MARKERSET [* / [markerset_file] {start_sym} {end_sym} {new_start#} {CLEAR}]
TEXTSET [* / [textset_file] {start_sym} {end_sym} {new_start#} {CLEAR}]
```

ARCPLOT only:

```
SHADESET [* / [shadeset_file] {start_sym} {end_sym} {new_start#} {CLEAR}]
```

With version 4.0, symbols can be loaded into ARCPLOT as the entire set of symbols in a symbol table or as independent symbol numbers from specified symbol tables. This allows you to build a custom symbol table from symbols contained in separate symbol tables.

* - Enter interactive mode.

lineset_file - the lineset file to load.
markerset_file - the markerset file to load.
shadeset_file - the shadeset file to load.
textset_file - the textset file to load.

{start_sym} - the first symbol number to load. The default is one.

{end_sym} - The last symbol number to load. The range of symbol numbers from {start_sym} to {end_sym} will be loaded.

{new_start#} - the symbol number to be assigned to {start_sym} when it is loaded. The second symbol to be loaded will be numbered {new_start#}+1, the third will be numbered {new_start#}+2 etc.

{CLEAR} - clears all symbols from [symbol_file] before loading the specified symbol numbers.

Examples:

symbolset symbol_file Loads symbols 1 to 255 from xxx

symbolset symbol_file 5 10 1 Loads symbols 5 to 10 from xxx, as numbers 1 through 6.
Symbol numbers 7 through 255 remain unchanged.

symbolset symbol_file 5 10 1 clear Clears all symbols from xxx and reload symbols 5 to 10 as
numbers 1 through 6.

A practical example would be the case where you have a line feature with 20 unique attribute values (numbered 1 to 20). Values 1 to 10 are to be displayed using symbols 31 to 40 from lineset file_one. Values 11 to 20 are to be displayed using symbols 63 to 72 from lineset file_two. Instead of making a lookup table and using selection sets to draw the arcs with their appropriate symbols, you can create a lineset file that accomplishes the same result.

```
LINESET file_one 31 40 1 CLEAR
```

```
LINESET file_two 63 72 11
```

```
ARCLINES coverage_name item_name
```

Optionally, the newly defined lineset file can be saved for future use with the new LINESAVE command. See below.

Refer also to the SYMPICK and SYMEDIT commands that allow interactive graphic symbol editing.

Saving symbol files: LINESAVE / MARKERSAVE / TEXTSAVE / SHADESAVE

New commands allow symbol files modified in an ARCEDIT or ARCPLOT session to be permanently saved.

```
LINESAVE [lineset_file]
```

```
MARKERSAVE [markerset_file]
```

```
TEXTSAVE [textset_file]
```

ARCPLOT only:

```
SHADESAVE [shadeset_file]
```

SHOW LINEINFO / MARKERINFO / TEXTINFO / SHADEINFO are new commands

```
SHOW LINEINFO [symbol] [layer] (8)
```

SHOW MARKERINFO [symbol] {4}
SHOW SHADEINFO [symbol] {17} (ARCPLOT only)
SHOW TEXTINFO [symbol] {9}

IMAGE support in ARCEDIT and ARCPLOT

Background images can be displayed in ARCEDIT and ARCPLOT. Currently, images must be BMP files and include the extension .BMP in their names. Each image must also have a world coordinate system file associated with it. This file must be named with the same prefix as the image file it represents, include the suffix .BPW and be located in the same directory as its image file. A world coordinate file is an ASCII file containing six lines with one number on each line. All numbers are in map units:

- cell size in the x direction (this is usually a positive number)
- rotation (currently, it must be zero)
- skew (currently, it must be zero)
- cell size in the y direction (this is usually a negative number)
- x coordinate of the center of the upper left pixel of the image
- y coordinate of the center of the upper left pixel of the image

This world file is the same format world file that is used in ArcView. For example if you have a .TIF image file with a corresponding .TFW, you could convert the .TIF to a .BMP and rename the .TFW to .BPW. The same thing applies to .JPG and .JGW files.

IMAGES in ARCEDIT

To work with an image,

- 1) Set an appropriate MAPEXTENT using a Coverage or an Image
- 2) Include Images in the background draw environment: BACKENVIRONMENT IMAGE ON
- 3) Add the Image to the current image list: IMAGE *image_name* (if necessary, remove unwanted images from the list: REMOVEIMAGE *image_name*)
- 4) Display the Image with the DRAW command

These commands support images in ARCEDIT:

MAPEXTENT [IMAGE [image...image]]
Sets the MAPEXTENT to the specified images.

BACKENVIRONMENT {IMAGE {ON / OFF}}
Sets the background draw environment for images. ON includes images in the background drawing environment and OFF removes images from the background drawing environment. Note that BACKENVIRONMENT {ALL {ON / OFF}} also includes images.

IMAGE [image]
Adds an image to the background image list. [image] is the pathname to a valid image file. Up to four images may be named to the list. Set BACKENVIRONMENT IMAGE ON to include images in the draw environment. When DRAW is issued, images will be drawn in the order they were added to the list. The DRAW command draws images first, back coverages next and the current edit coverage last.

REMOVEIMAGE [image]
Removes an image from the background image list. Images are added to the list with the IMAGE command.

SHOW IMAGE [1-4] (1)
Lists the image pathname of the specified image number (1,2,3 or 4) from the image list. An

optional SML variable number can be specified. This will cause the image pathname to be assigned to that variable number. If {sml_var} is not specified, the response is listed in the text window.

SHOW BACKENVIRONMENT {6}

A 6th sml variable will be assigned the background environment setting for IMAGES. If {sml_vars} are not specified, the BACKENVIRONMENT settings are displayed in the text window. In this case, the setting for IMAGES is the sixth one in the list.

STATUS {ALL / DRAW}

The section of this report that displays information about the BACKENVIRONMENT now has information about IMAGES. It displays either IMAGE ON or IMAGE OFF.

IMAGES in ARCPLOT

To work with an image,

- 1) Set an appropriate MAPEXTENT using a Coverage or an Image
- 2) Use the IMAGE command to draw the image

These commands support images in ARCPLOT:

MAPEXTENT [IMAGE [image...image]]

Sets the MAPEXTENT to the specified images.

IMAGE [image]

Draws images on the screen. The image name may be a pathname. Any number of images may be drawn.

Use the SHOWDIB command to preview images

SHOWDIB {image_name}

SHOWDIB is available in ARC, ARCEDIT and ARCPLOT.

Highlights of ARCEDIT 4.0

Continual cursor tracking when digitizing

When digitizing in ARCEDIT, the position of the digitizer's cursor is now tracked on the screen. Also, when digitizing lines, the line will be drawn on the screen as it is being entered. Note that your digitizer must support stream mode for these features to work.

Annotation in ARCEDIT

SETARROW command now supports up to 5 position points.

Text can be positioned using up to 500 points.

PUT [cover] {subclass} - can write the current annotations to a particular annotation subclass of [cover]

GET [cover] {subclass} {level} - can get annotations from a particular annotation subclass and level of [cover]

Annotation subclasses

At version 3.5.x, annotation could optionally be grouped and stored in levels. PC ARC/INFO 4.0 introduces another method of grouping annotation with the use of subclasses. Coverage annotation can now first be organized into subclasses and each subclass organized into different levels. Coverages may have any number of annotation subclasses, but these organizational groupings are still optional. The original unnamed or default annotation remains unchanged.

For example, a coverage named HIGHWAYS might have some general coverage annotation that is stored in the default class and two types of special annotation that can now be organized into the subclass names ROADS and TOWNS. Let's look at how this annotation is stored. All annotation is stored in TXT files. The default annotation is stored in the coverage directory as TXT.ADF and the subclass annotation in files named {subclass_name}.TXT. All annotation is drawn using the symbols and text stored in these coverage files. The coordinate array of each annotation is stored in TXX files.

Coverage HIGHWAYS would include these annotation files:

TXT.ADF (default)	ROADS.TXT	TOWNS.TXT
TXX.ADF	ROADS.TXX	TOWNS.TXX

If desired, each of these annotation subclasses can then be further grouped by annotation level. For example, certain annotation could be grouped by level to identify at which drawing scale the names are to appear on the map.

All commands that support annotation, now support annotation subclasses. This includes coverage maintenance commands such as COPYCOV, DESCRIBE and EXPORT. Copying a coverage with annotation subclasses to single precision will include the 'shifted' subclass files even though they cannot be used with PC ARC/INFO 3.5.x. In this way, subclass information is not lost when the coverage is copied back to double precision. In all cases, if a subclass is not named, the DEFAULT annotation is used.

EDITFEATURE [NONE / ARC / NODE / LABEL / TIC / ANNO{.subclass}] {Y}

The {.subclass} option lets you specify which annotation subclass to work with for the current edit coverage. If none is specified then the default annotation is used. If {.subclass} does not exist, it is created. Each EDITCOVERAGE can have only one subclass as the EDITFEATURE at a time. Changes made to one {.subclass} must be saved before switching to another {.subclass}, if you want to keep them. You will be reminded to save your changes before a new annotation subclass can be established or you can use the {Y} option to skip the prompt. SAVEFEAT is a new command that can be used to save annotation edits when switching between subclasses.

SAVEFEATURE - a new command

SAVEFEATURE saves changes made to the current edit feature of the current edit coverage. Use it when editing several different feature classes in a coverage and you want to control which changes are to be saved. SAVEFEATURE is particularly important when switching between annotation subclasses during an editing session, since you can only edit one annotation subclass at a time.

DRAWENVIRONMENT and BACKENVIRONMENT

The drawing environments for the EDITCOVERAGE and the BACKCOVERAGE use the same subclass as the current EDITFEATURE of EDITCOVERAGE.

SHOW ANNOSUBCLASS [0-4] {1}

This command returns the annotation subclass for a particular edit coverage. [0] returns the annotation subclass for the current edit coverage. By returning the subclass of an edit coverage [0-4] to an SML variable {1} you can restore an annotation subclass for editing after switching edit coverages or edit features.

EDITCOVERAGE roads

EDITFEATURE anno.name

&REM Change the edit coverage and set the edit feature to arc.

EDITCOVERAGE parcels

EDITFEATURE arc

&REM Retrieve the roads subclass name to SML variable 1.

SHOW ANNOSUBCLASS 1 1

&REM Change the edit coverage back to roads and reset the subclass edit feature

EDITCOVERAGE roads

EDITFEATURE ANNO.%1

LISTANNO and RENANNO

These ARC External commands can be used in ARCEDIT to list and rename annotation and annotation subclasses.

Stacked annotation in ARCEDIT

Annotation text can be "stacked" when it is added to a coverage. A backslash (\) within the string indicates a carriage return and causes a new line to be started. To keep a backslash as part of the annotation string, use two slashes next to each other. The alignment of stacked text can be established with the ANNOALIGNMENT command.

ANNOALIGNMENT establishes how blocks of text or stacked annotation will be drawn

ANNOALIGNMENT [LEFT / CENTER / RIGHT / AUTOMATIC]

[LEFT] - The text will be drawn left justified (default)

[RIGHT] - The text will be drawn right justified.

[CENTER] - The text will be drawn centered.

[AUTOMATIC] - text will be left, right or center justified based on the position of the text insertion point:

i.e., LL,CL,UL = Left justified

LR,CR,UR = Right justified

LC,CC,UC = Centered

New annotation pseudo items

Pseudo items can be used to alter the characteristics of existing annotation features. They are also useful for selecting and grouping annotation. Annotation characteristics are established when the features are added to a coverage. The parameters of each pseudo item match their 'host' command.

\$ALIGN - modifies the text alignment of stacked annotation. Selected annotation are immediately repositioned when this pseudo item is updated (and then drawn). \$ALIGN can have four different values (LEFT, RIGHT, CENTER and AUTOMATIC). Use MOVEITEM to assign values. If you try to assign any other values to \$ALIGN it will default to LEFT. Alignment is set with the ANNOALIGNMENT command.

\$FIT is used with two point text (ANNOTYPE POINT2). This is also set with the ANNOFIT command. It can be used to change whether annotation will fit between the two points. \$FIT can have two different values. Use MOVEITEM to assign values. If you try to assign any other values to \$FIT it will default to OFF.

\$JUSTIFY is used to reposition annotation around a feature or shape points. Selected annotation are immediately repositioned when this pseudo item is updated (and then drawn). \$JUSTIFY can have nine different values. Use MOVEITEM to assign values. If you try to assign any other values to \$JUSTIFY it will default to LL. This is set with the ANNOPOSITION command.

\$TEXT is used to assign a new text string to selected annotation features or can be used to select annotation which contains the specified text string.

Although not new, the pseudo item \$ID can now be used with annotation. For example, to group some annotation by ID number you could do the following:

```
EDITFEATURE ANNO  
SELECT MANY  
CALCULATE $ID = 1
```

New label-point pseudo items

The following pseudo items work with labels. They allow you to change the angle and scale of labels. These can be drawn in ARCPLOT and ARCEDIT.

\$ANGLE stores the amount of rotation of the text displayed with the label. The default \$ANGLE is zero degrees, which means that the marker symbol and text will not be rotated. Use CALCULATE to assign a value to \$ANGLE.

\$SCALE stores the scale factor for the drawing size of label point marker symbols. The size of label points in the Markerset file is multiplied by the value of \$SCALE to calculate a drawing size. Note that if the Markersize is 0.0, there will not be any apparent change in the size of the marker when it is drawn. The default value for \$SCALE is 1.0.

Arc and Label attributes can be written and read with the PUT and GET commands

PUT - If the PUT [cover] has no attributes or its attributes match the EDITCOVERAGE, feature attributes will be written to the PUT coverage. Otherwise, only coordinate information and the User-ID for the selected features are copied to [cover].

GET - If the EDITCOVERAGE has no attributes or its attributes match the GET cover, feature attributes will be copied from the GET [cover] to the EDITCOVERAGE. Otherwise, only coordinate information and the User-ID for the selected features are copied.

ARCPLOT commands can be used in ARCEDIT

ARCPLOT drawing and feature selection tools are now available for use during an ARCEDIT session. You can choose to "record" what is performed by ARCPLOT, so what you do is remembered during the ARCEDIT session, or you can access ARCPLOT tools as independent events.

There are two ways of using ARCPLOT in ARCEDIT. One uses the ARCPLOT command (in which ARCPLOT commands are only executed when entered) and the other uses the AP

command in which ARCPLOT commands are not only executed when they are entered but are also written to a file that is kept with the ARCEDIT drawing environment.

ARCPLOT {sml_file {sml_parameters}}

Starts an ARCPLOT session, but does not alter the current ARCEDIT screen display, graphic environment, current mapextent or position. All ARCPLOT commands can be run except DISPLAY. Note, though, that using MAPEXTENT, MAPPOSITION, MAPLIMITS, PAGESIZE, PAGEUNITS, LINESET, TEXTSET, MARKERSET and other commands that affect the graphics environment will not affect ARCEDIT when you return. Entering QUIT returns you to ARCEDIT. The graphics that were drawn in ARCPLOT remain on the screen.

AP {smlfile {sml parameters...}}
AP OFF

The AP command also lets you use ARCPLOT commands from within ARCEDIT. It, too, starts ARCPLOT and passes the current graphic environment (i.e., the graphic window, all graphics in that window, the current map extent, the current map position etc) to ARCPLOT. ARCPLOT commands can then be used to draw ARCPLOT symbology to the graphic window. When the QUIT command is executed, the program returns to ARCEDIT without clearing the screen. This time, though, the commands entered during the AP session, become part of the ARCEDIT drawing environment so each time the DRAW command is issued, the AP drawing elements are included in the display. The new drawing order for DRAW is:

- Images
- AP commands (if applicable)
- Backcoverages drawenvironment
- Editcoverage drawenvironment

To turn AP drawing off, use the AP OFF command.

Remember that the AP command “records” the keystrokes you make while it is active. Therefore, if you use an interactive ARCPLOT tool (such as feature selection using the mouse), you will similarly be prompted for input when the AP elements are re-issued by the ARCEDIT DRAW command. Note that none of the SML commands used during an AP session are recorded. In this way you can get file and coordinate information using these tools without being prompted again for it when you return to your ARCEDIT session. This means that activities carried out using &WIN would not be recorded, but those performed using the WIN command would be repeated every time a DRAW was issued. For example, to get coordinates interactively for use with the SELECT BOX option of ARCPLOT, you can enter the following:

```
&GETXY 1 2 9  
&GETXY 3 4 9 2  
SELECT cover BOX %1 %2 %3 %4
```

Where variable 1 and 2 contain the co-ordinates of one corner of the box and variables 3 and 4 hold the co-ordinates of the other corner. Variable 9 contains the key value used for the selection and the 2 option of the second &GETXY requests that a box is rubber-banded from the current (first) XY, location to wherever the mouse is moved.

An SML file can be specified if you want your ARCPLOT commands automated. Processing will return to ARCEDIT when QUIT or &RETURN is encountered.

A new ARCEDIT DRAW file is written each time AP is used. You cannot append to or edit the file once it is written.

As with the ARCPLOT command, using MAPEXTENT, MAPPOSITION, MAPLIMITS, PAGESIZE, PAGEUNITS, LINESET, TEXTSET, MARKERSET and other commands that affect the graphics environment will affect ARCPLOT, but it will not affect ARCEDIT when you return.

It is important to note that any changes made with the &WIN command using ARCPLOT or AP (dialog boxes, menus, patterns, colors) WILL affect ARCEDIT when you return. It is as if you executed the &WIN command in ARCEDIT.

Highlights of ARCPLOT 4.0

Enhanced printing support in ARCPLOT

When PAGESIZE is altered in ARCPLOT, the same change is applied to %pagesize of the Windows Printer Options.

Feature on Feature selection in ARCPLOT

The selection commands in ARCPLOT have a new OVERLAP option that permits the features of one coverage to be selected using the selected features of another coverage.

Feature selection based on the location of a point is now supported

Features can also be selected based on a new POINT option with ARCPLOT selection commands. Features that are near a specified point location may be added to the selected set. A circle, with a radius of SEARCHTOLERANCE is defined around the POINT. All features that fall within or pass through the area defined by that circle may be selected.

Defining lines and polygon outlines in ARCPLOT with such commands as LINE and SHADE allow additional controls such as splining, squaring and undoing the last point

Interactively entering a line or polygon outline for line drawing, shade drawing or for feature selection includes options for splining the line through a point and squaring.

Making your own fonts, characters and marker symbols

FONTEDIT was used by previous PC ARC/INFO versions to create and edit text and marker symbols. Since it was a DOS program, it no longer exists. In its place you can use any 3rd party Windows font editor to edit or create TrueType fonts and any bitmap editor to create character and symbol markers. PC ARC/INFO 4.0 can access all available Windows fonts as marker symbols.

Using bitmap images as markers in ARCPLOT and ARCEDIT

Version 4.0 allows bitmap images to be used as non-scalable marker symbols. Non-scalable markers are marker symbols that do not change size when the scale of the map is altered by changing the Map Extent or by hardware zooming.

Bitmaps that are to be used as markers must be saved in the directory specified by WIN PATH or in the ARCEXE\SYMBOLS directory and must be named 'MRKnnnnn.bmp' (where 'nnnnn' is a number 00128-32767 and one that includes all leading zeros).

Note that numbers 00128 through 00143 (MRK00128.BMP etc.) currently contain bitmap versions of the IGL marker patterns 0 to 15. You may keep these for your use or you may choose to delete them from your system. They are approximately 0.122 inches in height.

To draw a saved bitmap in ARC PLOT using its existing size and color(s), set the following parameters:

- 1) MARKERFONT 0 - the Font must always be '0' to use bitmap markers
- 2) MARKERSIZE 0 - a size of '0' causes the bitmap to be drawn at its existing size

Using a marker size other than 0 will draw the bitmap at the specified size. Marker size is always in Page Units and is measured as the marker's height. Expanding or compressing the existing size of a marker may cause the output to become distorted. You can always edit the existing size of a bitmap using a 3rd party bitmap editor.

- 3) MARKERCOLOR 0 - this causes the existing colors in the bitmap to be used

Using the color '0' allows you to use small multicolored pictures and symbols for markers. Everything underneath the placed bitmap will be covered over. If MARKERCOLOR is set to a nonzero value, the bitmap is assumed to be a black and white image. The 'nonwhite' parts of the image will be drawn with the specified marker color. The 'white' parts of the image will be treated as transparent and will not change the underlying graphics.

- 4) MARKERPATTERN *nnnnn* - specifies that the bitmap file MRK*nnnnn*.bmp is to be drawn. It is first looked for on WIN PATH and, if not found there, is looked for on the ARCEXE\SYMBOLS directory.

Note that the leading zeros of the pattern number are not required on the MARKERPATTERN command line. If the specified pattern number does not exist, MARKERPATTERN is set to 0.

- 5) MARKER * (or other marker drawing command) - draw the bitmap at the location of the cursor

To draw the same saved bitmap in ARCEDIT, use the MARKERSET * command to interactively enter the marker characteristics listed above. Then use the DRAW command. The drawenvironment must include LABELS.

More importantly, for the labels to be drawn, the value of pseudo item \$SYMBOL must match the symbol number specified for the bitmap image with the MARKERSET * dialog.

By default, markers in ARCEDIT have pseudo item \$SYMBOL = 33. (When, selected, they are drawn with the value specified by SETDRAWSYMBOL.) Either set the bitmap to Symbol 33 or CALCULATE the \$SYMBOL values of the features to the bitmap marker symbol number.

For example:

MARKERSET *

Symbol: **1**

Color: **0**

Pattern: *nnn*

Font: **0**

Size: **0**

SELECT *

1 element(s) new selected

CALCULATE \$SYMBOL = 1
UNSELECT ALL
DRAW

Notes:

The marker values mentioned above can be put in a marker symbol table using the SIZE, COLOR and PATTERN items. In this way you can have bitmap markers in your symbol tables.

The non-scalable vector markers from previous versions of PC ARC/INFO still exist with version 4.0. They are accessed by specifying a marker font of 0, a marker size of 0 and a pattern number from 0 through 15. They are drawn with the current marker color.

Size and position of drawing elements can be interactively altered

These commands now support a cursor option (*) to allow you to specify their location or their size:

KEYBOX	KEYSEPARATION	LINESIZE	MARKERSIZE
OVERAREA	OVERSEPARATION	TEXTSIZE	

MAP compositions can be opened and drawn on top of existing graphics

MAP [map_name / END] {DRAW / NODRAW / NOCLEAR}

Use the NOCLEAR option to draw a map composition without first clearing the screen. Existing graphics are left on the screen and the map composition is drawn on top of them. This lets you draw images and not lose the image when you use the MAP command. The DRAW option clears the screen before drawing the map composition. NODRAW does not clear the screen or draw map composition elements.

SHOW LINESET / MARKERSET / SHADESET / TEXTSET are new in ARCPLOT

SHOW LINESET {1}
SHOW MARKERSET {1}
SHOW SHADESET {1}
SHOW TEXTSET {1}

Note that these SHOW commands are still available in ARCEDIT.

New ARCPLOT annotation and text features

Text can be positioned using up to 500 points.

All commands that display annotation support the new annotation subclass feature.

Stacked and concatenated text strings in ARCPLOT

The following commands have a new parameter named [text_expression] that replaces the previous parameter [item]. Applications that use the following commands do not have to be modified since this is an enhancement to the usage of the commands and not a direct change:

ARCTEXT	DROPLINE	ITEMTEXT	LABELTEXT
POINTTEXT	POLYGONTEXT	TEXT	TEXTFILE

For example: POLYGONTEXT [cover] [text_expression] {lookup_table}

[text_expression] has the following usage:

[item / text] { [operator] [item / text] } ... { [operator] [item / text] }

item - any item, pseudo item, combined item or related item. Trailing blanks are always removed during concatenation.

text - any text string. Text must always be surrounded with matching single or double quotes. To include a quotation mark in the text, use two quotation characters together ("").

operator - one of two characters. Use the vertical bar (|) for concatenation or the 'backslash (\) for a new line. There should be at least one blank space on either side of the operator. To include a backslash in the text, use two backslash characters together (\\). Up to 20 operators may be specified per command line.

This allows multiple items to be displayed in the same text string or stacked in a block. The alignment of stacked text can be established with the TEXTALIGNMENT command. (See below.)

The total length of the stacked and/or concatenated text cannot exceed 256 characters.

Typeset commands can be included in the text expression by first issuing the TEXTSTYLE TYPESET command. For example, the following command sequence will underline the value of *item*:

```
TEXTSTYLE TYPESET
LABELTEXT cover item | '!und;'
```

Typeset commands can be included in [text_expression] by first issuing the TEXTSTYLE TYPESET command. For example, the following commands will underline the value of *item*:

```
TEXTSTYLE TYPESET
LABELTEXT cover item | '!und;'
```

TEXTALIGNMENT - new command in ARCPLOT

TEXTALIGNMENT [LEFT / CENTER / RIGHT / AUTOMATIC]

Aligns blocks of text or stacked annotation.

[LEFT] - The text will be drawn left justified (default)

[RIGHT] - The text will be drawn right justified.

[CENTER] - The text will be drawn centered.

[AUTOMATIC] - text will be left, right or center justified based on the position of the text insertion point:

LL,CL,UL - Left justified

LR,CR,UR - Right justified

LC,CC,UC - Centered

The following commands use the TEXTALIGNMENT setting when displaying stacked annotation:

ARCTEXT	DROPLINE	ITEMTEXT	LABELTEXT
POINTTEXT	POLYGONTEXT	TEXT	TEXTFILE

END OF FILE